

Este proyecto nace de la necesidad de automatizar el estudio de sistemas hidráulicos y de control de roturas en presas.

Para realizar el estudio de sistemas hidráulicos se usarán un número indeterminado de sensores de nivel, presión y caudal. El número de sensores que se pueden utilizar viene determinado por el material disponible.

Estos sensores se conectarán a unas tarjetas de National Instruments modelo NI 9208 y éstas a su vez a un chasis modelo CompactDAQ NI-9174 con cuatro ranuras. Conectando este chasis al ordenador podremos obtener los datos provenientes de los sensores. También se podrá controlar una válvula para determinar la cantidad de agua que fluye en nuestro experimento. Esta válvula está conectada a una tarjeta NI-9264 que se conectará al chasis en su última posición

Para detectar y estudiar posibles roturas en presas se dispone de un motor y un láser con los cuales se puede barrer la superficie de una presa y obtener una imagen en tres dimensiones de la misma procesando los datos provenientes del laser.

Para recoger los datos de los sensores y controlar una válvula se ha desarrollado una aplicación utilizando LabVIEW, un programa creado por National Instruments.

Para poder controlar el motor y el láser se parte de una aplicación que ya estaba realizada en LabVIEW. El objetivo ha sido detectar y corregir una serie de errores en la misma.

Dentro del proyecto, además de la explicación detallada de la aplicación para los sensores y la válvula, y las pruebas realizadas para detectar y corregir los errores de la aplicación del láser y el motor, existe: una breve introducción a la programación en LabVIEW, la descripción de los pasos realizados para el conexionado de los sensores con las tarjetas, los manuales de usuario de las aplicaciones y la descripción de los equipos utilizados.

This project starts from the need to automate the study of hydraulic systems and control dam breaks.

For the study of hydraulic systems it will be used an unspecified number of level, pressure and flow sensors. The number of sensors that can be used is determined by the available material.

These sensors are connected to a NI 9208 National Instruments target and these cards to a NI-9174 CompactDAQ chassis with four slots. Connecting the chassis to a computer we will obtain data from the sensors. We also can control a valve to determine the amount of water flowing in our experiment. This valve is connected to a NI-9264 card and this card to the last position of the chassis.

To detect and study dams breakage it used a motor and a laser. With these two devices we can scan the surface of a dam and obtain a three-dimensional image processing data from the laser.

To collect data from the sensors and control the valve it has developed an application using LabVIEW, a program created by National Instruments.

To control the motor and the laser it is used an application that was already created using LabVIEW. The aim of this part has been detect and correct a number of errors in this application.

Within the project, in addition to the detailed explanation of the application for sensors and valve, and tests to detect and correct errors in the application of lasers and the motor, there is: a brief introduction to programming in LabVIEW, the description of the steps taken for connecting the sensors with cards, user manuals and application description of the equipment used.



## PROYECTO FIN DE CARRERA PLAN 2000

E.U.I.T. TELECOMUNICACIÓN

**TEMA:** Instrumentación Virtual (IV).

**TÍTULO:** Sistemas de control de roturas en presas y caracterización de sistemas hidráulicos.

**AUTOR:** Luis Jerónimo Moya.

**TUTOR:** Eduardo Barrera López de Turiso.

**Vº Bº.**

**DEPARTAMENTO:** SEC



**Miembros del Tribunal Calificador:**

**PRESIDENTE:** Magdalena González Martín.

**VOCAL:** Eduardo Barrera López de Turiso.

**VOCAL SECRETARIO:** Sergio López Gregorio

**DIRECTOR:**

**Fecha de lectura:** 28 de septiembre de 2012

**Calificación:**

**El Secretario,**

### RESUMEN DEL PROYECTO:

Con el objetivo de estudiar sistemas hidráulicos como presas, canales o vertederos hidráulicos se pretende diseñar una aplicación que permita recoger los datos provenientes de unos sensores de presión, nivel y caudal, y controlar el funcionamiento de una válvula.

La aplicación ha de ser flexible ya que el número de sensores puede variar en función del experimento. El número máximo de sensores que se pueden utilizar viene fijado por el material disponible.

También se deberá buscar y corregir los errores de una aplicación que permite controlar un motor y un láser usados para detectar roturas en presas.

Para todo lo anterior se utilizará LabVIEW, un programa de National Instruments, que nos permitirá crear aplicaciones para la adquisición de datos y de control.

**ESCUELA UNIVERSITARIA DE INGENIERIA  
TECNICA DE TELECOMUNICACION**



**PROYECTO FIN DE CARRERA**

**SISTEMAS DE CONTROL DE ROTURAS EN  
PRESAS Y CARACTERIZACIÓN DE SISTEMAS  
HIDRÁULICOS**

**Septiembre, 2012**

**Luis Jerónimo Moya**

## Agradecimientos

En esta parte de proyecto me gustaría agradecer a todas esas personas que han estado cerca de mí, que me han apoyado y soportado durante los años que he estudiado esta carrera.

*A mis padres, por preocuparse tanto por mí durante mis cuatro años de carrera.*

*A Sara, por soportarme y apoyarme cuando las cosas no iban bien.*

*A mis compañeros de residencia, por hacer de estos últimos cuatro años los mejores de mi vida.*

*A mis amigos de universidad, con los que he compartido tantas comidas y horas de estudio.*

*A Eduardo, por confiar en mí y brindarme la oportunidad de adentrarme en el mundo de la instrumentación.*

*A todos los profesores que he tenido a lo largo de la carrera, por todo lo que he aprendido gracias a ellos.*

*A Hibber, Andrea, León y José Juan, por ayudarme en todo lo que he necesitado durante los tres meses que he trabajado con ellos.*

*Al soporte técnico de National Instruments, por ayudarme con todas mis dudas.*

*En general, gracias a todos vosotros y a los que no he podido nombrar.*

*“No hay mayor grandeza  
que vencerse a sí mismo.  
Esa es la grandeza...” Sócrates*

## INDICE

<b>1. INTRODUCCION.....</b>	<b>12</b>
1.1. Objetivos.....	13
1.2. Estructura de la memoria.....	14
<b>2. CONCEPTOS BÁSICOS.....</b>	<b>15</b>
2.1. LabVIEW.....	16
2.1.1. Introducción.....	16
2.1.2. Instrumentos Virtuales.....	17
2.1.2.1. Panel frontal.....	17
2.1.2.2. Diagrama de bloques.....	18
2.1.2.3. Icono-conector.....	18
2.1.2.4. Ventanas de panel y diagrama.....	19
2.1.2.4.1. Barras de herramientas del panel frontal y diagrama de bloques.....	19
2.1.2.4.2. Paletas.....	21
2.1.2.4.3. Nodos.....	23
2.1.2.4.4. Terminales.....	23
2.1.2.4.5. Cables.....	24
2.1.2.4.6. Flujo de datos.....	24
2.1.3. Desarrollo de aplicaciones de adquisición de datos con LabVIEW.....	25
2.1.3.1. Introducción.....	25
2.1.3.2. Comunicación serie.....	26
2.1.3.2.1. Instrumentos serie.....	26
2.1.3.3. Driver de un instrumento.....	27
2.1.3.3.1. Programación a bajo nivel VISA.....	28
<b>3. DESARROLLO DEL SISTEMA.....</b>	<b>29</b>
3.1. Adquisición de datos de sensores y control de una válvula.....	30
3.1.1. Material utilizado y conexiones.....	30
3.1.2. Distribución de canales por tarjeta.....	36
3.1.3. Base de datos: Documento de Excel.....	38
3.1.4. Aplicaciones.....	41
3.1.4.1. Aplicación para adquirir datos de sensores.....	41
3.1.4.2. Aplicación para adquirir datos de sensores y controlar una válvula.....	66
3.1.5. Manual de usuario.....	71
3.1.5.1. Panel frontal de la aplicación sin válvula.....	71

3.1.5.2. Panel frontal de la aplicación con válvula.....	78
3.1.6. Pruebas realizadas a sensores que se van a utilizar en experimentos.....	82
3.2. Láser y motor.....	86
3.2.1. Introducción.....	86
3.2.2. Aplicación.....	86
3.2.3. Detección y corrección de errores.....	91
3.2.3.1. Detección y corrección de errores de la aplicación.....	91
3.2.3.1.1. Memoria Virtual.....	92
3.2.3.1.2. Modificación de la aplicación.....	93
3.2.3.1.3. Compact RIO.....	97
3.2.3.2. Detección y corrección de error en la coordenada “Z”.....	99
<b>4. CONCLUSIONES.....</b>	<b>104</b>
<b>5. CARACTERISTICAS DE EQUIPOS UTILIZADOS.....</b>	<b>107</b>
5.1. Módulo de entradas en corriente NI-9028.....	108
5.2. Chasis NI <i>CompactDAQ</i> 9174 USB de 4 ranuras.....	111
5.3. Fuente de alimentación NI PS-25.....	113
5.4. Sensor de nivel ultrasónico modelo LU-05.....	115
5.5. Sensor de presión modelo XA-300.....	119
5.6. Sensor de presión modelo XA-700.....	122
5.7. Válvula AUMA MATIC AM 01.1 / AM 02.1.....	125
<b>6. BIBLIOGRAFÍA.....</b>	<b>128</b>



## INDICE DE FIGURAS

Figura 1: Bancos de pruebas.....	13
Figura 2: Panel frontal.....	17
Figura 3: Diagrama de bloques.....	18
Figura 4: Icono y su conector.....	18
Figura 5: Barra de herramientas del panel frontal.....	19
Figura 6: Barra de herramientas del diagrama de bloques.....	20
Figura 7: Paleta de herramientas.....	21
Figura 8: Paleta de controles.....	21
Figura 9: Paleta de funciones.....	22
Figura 10: Paleta de adquisición de datos <i>DAQmx</i> .....	25
Figura 11: Paleta de adquisición de datos <i>VISA</i> .....	28
Figura 12: Tarjeta NI-9208.....	30
Figura 13: NI cDAQ-9174 y CD de <i>drivers</i> .....	31
Figura 14: Fuente de alimentación PS-15.....	32
Figura 15: Conexión de la fuente de alimentación y sensores con la tarjeta NI-9208.....	32
Figura 16: Esquema de conexión de fuente de alimentación.....	33
Figura 17: Conexión realizado con fuente de alimentación PS-15.....	33
Figura 18: Clemas.....	34
Figura 19: Adaptador pines tarjeta NI-9208.....	34
Figura 20: Objeto para el conexión.....	35
Figura 21: Carcasa.....	38
Figura 22: Documento de Excel.....	40
Figura 23: Estructura <i>Flat Sequence</i> .....	41
Figura 24: Estructura <i>Stacked Sequence</i> .....	41
Figura 25: Función <i>Initialize Array</i> .....	42
Figura 26: <i>New report.vi</i> .....	42
Figura 27: <i>For Loop</i> .....	43
Figura 28: <i>Shift register</i> en bucle <i>for</i> .....	43
Figura 29: <i>busqueda sensor.vi</i> .....	44
Figura 30: <i>Excel Get Data.vi</i> .....	45
Figura 31: Función <i>Build Array</i> .....	45
Figura 32: Función <i>Array To Cluster</i> .....	46
Figura 33: <i>guardar datos.vi</i> .....	46
Figura 34: <i>Estructura Case</i> .....	48
Figura 35: Función <i>Insert Into Array</i> .....	49

Figura 36: Función <i>Decimal String To Number</i> .....	50
Figura 37: Función <i>Equal?</i> .....	51
Figura 38: <i>DAQmx Timing (Sample Clock).vi</i> .....	52
Figura 39: Bucle <i>while</i> .....	53
Figura 40: <i>DAQmx Read (Analog 1D DBL NChan 1Samp).vi</i> .....	54
Figura 41: calculos.vi y ajustes.vi.....	55
Figura 42: Función <i>Index Array</i> .....	56
Figura 43: <i>Formula Node</i> .....	57
Figura 44: Función <i>Array Subset</i> .....	60
Figura 45: <i>gestion_graficos.vi</i> .....	60
Figura 46: Función <i>Merge Signals</i> .....	61
Figura 47: Indicadores <i>Waveform Chart</i> .....	62
Figura 48: <i>Properties Waveform Chart</i> .....	62
Figura 49: <i>Active Plot</i> .....	63
Figura 50: <i>Visible</i> .....	63
Figura 51: <i>Write Characters To File.vi</i> .....	64
Figura 52: Función <i>Build Path</i> .....	64
Figura 53: Función <i>Concatenate Strings</i> .....	64
Figura 54: Función <i>Array To Spreadsheet String</i> .....	65
Figura 55: <i>DAQmx Clear Task.vi</i> .....	65
Figura 56: <i>Read From Spreadsheet File.vi</i> .....	67
Figura 57: <i>Función Get Date/Time In Seconds</i> .....	68
Figura 58: <i>PIDHIDER2.vi</i> .....	69
Figura 59: <i>DAQmx Write (Analog DBL 1Chan 1Samp).vi</i> .....	70
Figura 60: <i>Front Panel</i> aplicación sin control de válvula.....	71
Figura 61: Selector de la ruta de Documento de Excel.....	71
Figura 62: Control Muestras/Segundo/canal.....	72
Figura 63: Control Número de tarjetas (AI).....	72
Figura 64: Botón <i>Run</i> .....	72
Figura 65: Indicadores gráficos de nivel, presión, y caudal.....	73
Figura 66: Traza y opciones de la misma.....	73
Figura 67: Opciones para ver o no ver trazas.....	74
Figura 68: No ver.....	74
Figura 69: Ver.....	74
Figura 70: Traza oculta.....	74
Figura 71: Traza visible.....	74
Figura 72: Trazas.....	75

Figura 73: <i>SCROLL</i> .....	76
Figura 74: Nombre y ruta del archivo de registro de medidas.....	76
Figura 75: No capturar.....	76
Figura 76: Capturar.....	76
Figura 77: Indicador <i>array mediciones</i> .....	77
Figura 78: Indicador array mediciones explicado.....	77
Figura 79: Control <i>TERMINAR APLICACIÓN</i> .....	77
Figura 80: <i>Front Panel</i> aplicación con control de válvula.....	78
Figura 81: Control válvula activado.....	79
Figura 82: Control válvula desactivado.....	79
Figura 83: Controles para controlar la válvula.....	79
Figura 84: Control automático del caudal activado.....	80
Figura 85: Objetivo del control automático.....	80
Figura 86: Control <i>Q referencia</i> .....	80
Figura 87: Selección del documento para el control de la válvula.....	80
Figura 88: Control posición llave activado.....	80
Figura 89: Control abertura llave (°).....	81
Figura 90: Poner tiempo a cero.....	81
Figura 91: Pieza.....	82
Figura 92: Pieza con los sensores roscados.....	82
Figura 93: Sensores XA-700.....	83
Figura 94: Tubo para realizar pruebas.....	83
Figura 95: Láser y motor.....	86
Figura 96: Pasarela.....	86
Figura 97: <i>Event Structure</i> .....	87
Figura 98: <i>Front Panel</i> LaserCamino.vi.....	88
Figura 99: Pantalla azul con letras blancas.....	91
Figura 100: Mensajes de error.....	91
Figura 101: matrices_datos.vi.....	93
Figura 102: Función <i>Greater Or Equal?</i> .....	94
Figura 103: Función <i>Greater?</i> .....	95
Figura 104: Diagrama de bloques de matrices_datos.vi.....	96
Figura 105: <i>Shift register</i> y diagrama de bloques donde se ha realizado el cambio.....	96
Figura 106: <i>FPGA</i> con chasis.....	97
Figura 107: Controlador.....	97
Figura 108: Controlador unido a chasis y <i>FPGA</i> .....	97
Figura 109: Desnivel del suelo.....	100

Figura 110: Desnivel con aproximación lineal.....	101
Figura 111: Corrección del error.....	103
Figura 112: Tarjeta NI-9208.....	108
Figura 113: <i>CompactDAQ</i> 9174.....	111
Figura 114: Fuente de alimentación PS-15.....	113
Figura 115: Sensor de nivel LU-05.....	115
Figura 116: Cableado sensor LU-05.....	117
Figura 117: Dimensiones sensor LU-05.....	118
Figura 118: Transmisor de presión XA-300.....	119
Figura 119: Rosca transmisor XA-300.....	120
Figura 120: Dimensiones transmisor XA-300.....	121
Figura 121: Conexiones transmisor XA-300.....	121
Figura 122: Transmisor de presión XA-700.....	122
Figura 123: Rosca transmisor XA-700.....	123
Figura 124: Cuerpo transmisor XA-700.....	124
Figura 125: Conexiones eléctricas transmisor XA-700.....	124
Figura 126: Válvula AUMA.....	125
Figura 127: Controles válvula.....	127

## INDICE DE TABLAS

Tabla 1: Distribución por tarjeta para versión sin válvula.....	36
Tabla 2: Distribución por tarjeta para versión con válvula.....	37
Tabla 3: Orden de los datos característicos de un sensor en el <i>array</i> .....	47
Tabla 4: Operaciones para mantener el orden de posiciones.....	50
Tabla 5: Tamaño del <i>buffer</i> .....	53
Tabla 6: Ejemplo situación de trazas.....	75
Tabla 7: Media de errores sensores XA-300.....	84
Tabla 8: Media de errores sensores XA-700.....	85
Tabla 9: Valores asociados a cada resolución.....	95
Tabla 10: Relación altura corriente.....	110

# 1. INTRODUCCIÓN

## 1.1. Objetivos

El proyecto se ha desarrollado a través de una beca de tres meses de duración en la Escuela Técnica Superior de Ingeniería de Caminos, Canales y Puertos de la UPM.

Dentro de la Escuela de Caminos se ha utilizado el laboratorio de Hidráulica donde se encontraban todos los materiales utilizados. En este laboratorio están todos los bancos de pruebas que permitían simular presas, canales y otros sistemas hidráulicos que van a ser estudiados por medio de una serie de sensores que son colocados en lugares específicos. En la figura 1 podemos ver algunos de los bancos de prueba del laboratorio.



**Figura 1: Bancos de pruebas**

La mayoría de los objetivos fueron fijados inicialmente y otros se han ido incorporando. Los objetivos son los siguientes:

- Desarrollo de una aplicación en LabVIEW para recoger los datos provenientes de sensores de presión, caudal y nivel con el objetivo de estudiar sistemas hidráulicos. La aplicación debe ser capaz de adaptarse a un número de sensores que puede variar en función del experimento que se vaya a realizar. El número máximo de sensores viene determinado por el material disponible.
- Incorporación a la aplicación anterior de parte del código desarrollado en otra aplicación para el control de una válvula.
- Detección y corrección de errores de un programa que permite controlar un motor y un láser.
- Corrección de un error existente en el láser en relación al eje Z de coordenadas.

## 1.2 Estructura de la memoria

La memoria tiene 3 partes claramente diferenciadas:

- En la primera parte se explican los conceptos básicos que el lector debe conocer para que así sea más asequible la lectura del proyecto.
- La segunda parte del proyecto se corresponde con la descripción de todas las tareas realizadas, empezando con el desarrollo de una aplicación para adquirir datos de sensores para controlar una válvula y finalizando con la el proceso seguido para intentar detectar y corregir los errores en una aplicación que controla un motor y un láser.
- La última parte del proyecto incluye las conclusiones del mismo y un apartado en el que se describen de las características de los equipos que han sido utilizados.



## **2. CONCEPTOS BASICOS**

## 2.1 LabVIEW

### 2.1.1. Introducción

LabVIEW es una herramienta software desarrollada por *National Instruments* para la creación de aplicaciones de adquisición de datos y de control. LabVIEW es una herramienta de programación gráfica que contiene todas las utilidades necesarias para crear, depurar y generar programas ejecutables de aplicación al mundo de la automatización industrial y del test y medida.

Las aplicaciones desarrolladas en LabVIEW tienen una estructura basada en la utilización de los siguientes elementos:

- Interfaz de usuario: los objetos del interfaz de usuario permiten elaborar paneles interactivos de una forma rápida. Los diferentes elementos disponibles en la interfaz de usuario proporcionan mecanismos para leer y visualizar datos mediante menús, paneles, controles y cajas de diálogo.
- Diagrama de bloques: es el código de la aplicación en LabVIEW y está formado por estructuras de control (*while*, *case*, etc), terminales que identifican elementos gráficos de E/S, y por nodos y funciones que realizan operaciones con los datos o controles de entrada para representarlos sobre los indicadores.
- Adquisición de datos: en esta parte del programa se realiza el control de los diferentes elementos de adquisición de datos, estos podrían ser bien sistemas de bus externo, como instrumentos programables mediante las interfaces RS232, GPIB, VXI, PXI, bien de bus interno, como tarjetas de adquisición conectadas en los buses de expansión (PCI) y otros elementos como autómatas, sistemas de control de movimiento, etc.

LabVIEW facilita la utilización de este tipo de sistemas mediante un conjunto de librerías específicas:

- Data Acquisition (NIDAQ y DAQmx)*: librerías para el control de tarjetas de adquisición de datos. La librería ofrece la posibilidad de gestionar las E/S analógicas y digitales mediante las técnicas más habituales como son el “pooling”, interrupciones, DMA y bus máster en arquitecturas.
- VISA (Virtual Interface Software Architecture)*: librería de control de instrumentos que permiten gestionar cualquiera de las interfaces hardware habituales como PXI, IEEE488, VXI Y RS232 que se utilizan en las aplicaciones de test y medida. Es una librería que se ha especificado dentro del estándar VXI Plug&Play.

### 2.1.2 Instrumentos Virtuales

Los programas desarrollados en LabVIEW se conocen por el nombre de VIs (*Virtual Instruments* o Instrumentos Virtuales). Los VIs tienen tres partes principales: el panel frontal, el diagrama de bloques y el icono-conector.

#### 2.1.2.1 Panel Frontal

El panel frontal (figura 2) es una ventana donde el programador podrá colocar los diferentes elementos con los que interactuará. Entre ellos existirán botones, palancas, visualizadores números, gráficas, etc. Estos elementos de interfaz se clasifican en controles e indicadores dependiendo si se utilizan para introducir o visualizar información de la aplicación.

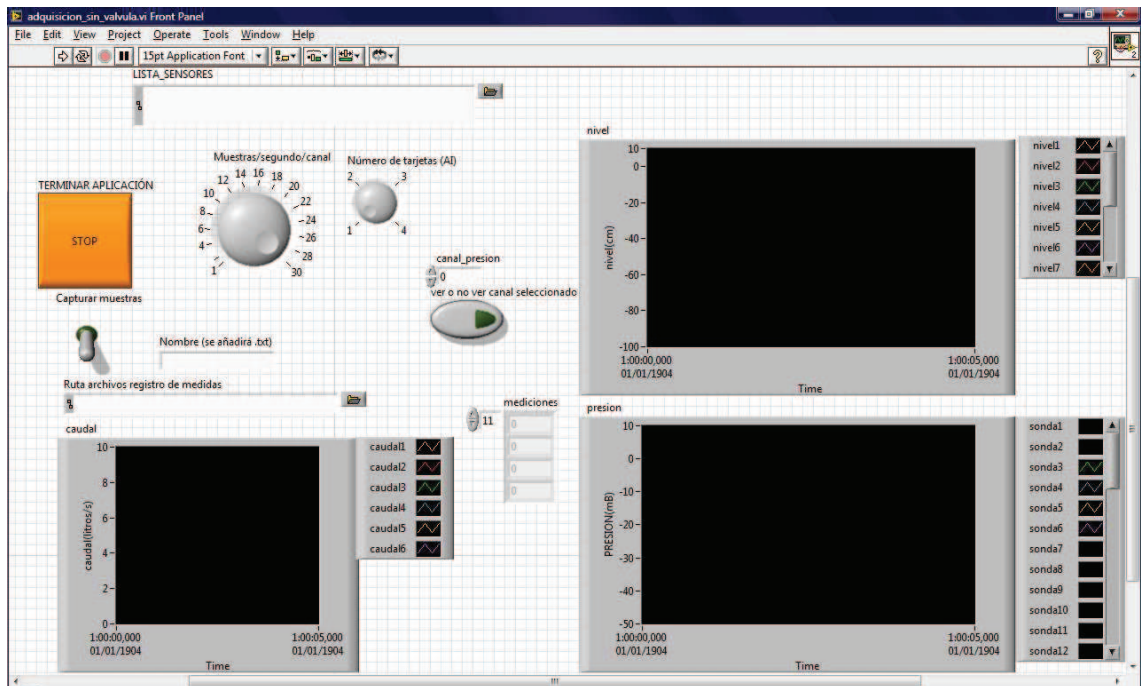


Figura 2: Panel frontal

### 2.1.2.2 Diagrama de bloques

El código fuente de una aplicación desarrollada en LabVIEW se especifica en lo que se denomina el diagrama de bloques (figura 3). Los controles e indicadores del panel frontal aparecen representados en el diagrama de bloques del VI mediante unos recuadros que se denominan “terminales”.

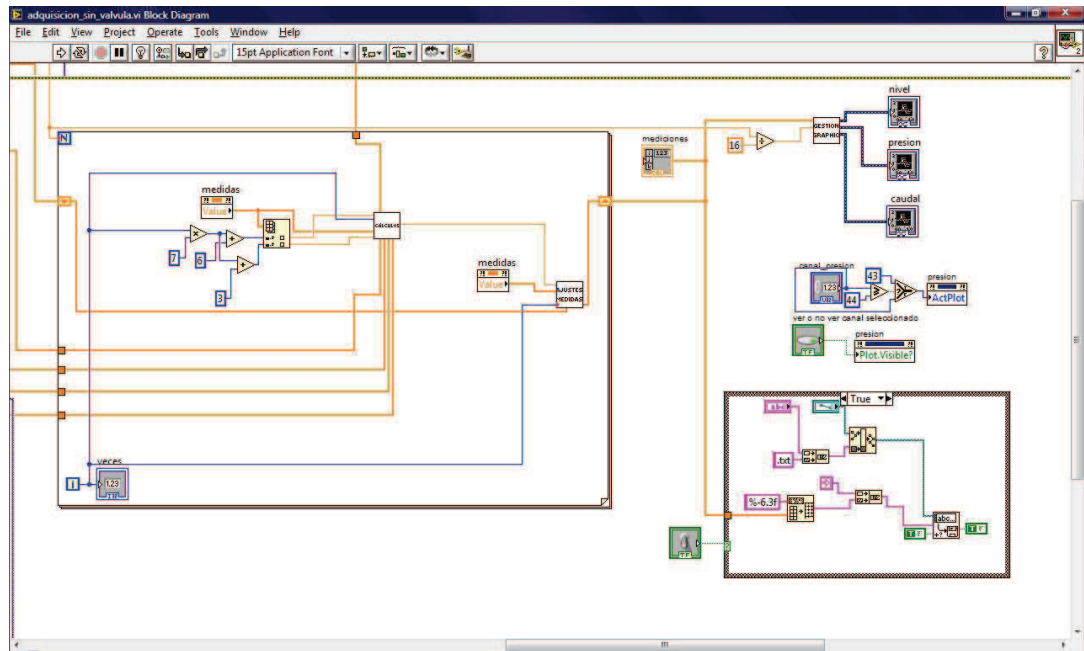


Figura 3: Diagrama de bloques

### 2.1.2.3 Icono y conector

La parte que queda por representar de un VI es su icono. Este es imprescindible para poder utilizarlo dentro del diagrama de bloques de otro VI más complejo. Tiene la misma misión que una función usada en lenguaje C, se define una función para utilizarla en otros programas o zonas del mismo con el fin de no reescribir el código varias veces.

Una vez el VI queda representado por un icono, éste se podrá conectar a otros objetos del diagrama de bloques de otro VI. El conector determina dónde tiene que ir los terminales de entrada y salida de datos en el icono. En la figura 4 se puede ver un ejemplo de icono con su conector.

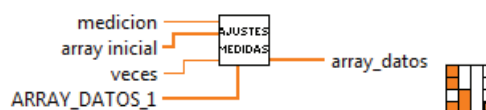


Figura 4: Icono y su conector

La principal característica que hace importante a LabVIEW radica en la naturaleza jerárquica que ofrecen sus programas, siendo posible utilizar el número de VIs que queramos dentro del diagrama de bloques de otro VI. Esto supone una gran ayuda para el usuario de cualquier Instrumento Virtual, al que le resultará mucho más fácil realizar, depurar o entender un programa. Un VI que se utiliza como subrutina en el diagrama de bloques de otro VI recibe el nombre de “subVI”.

#### 2.1.2.4 Ventanas de panel y diagrama.

Al iniciar LabVIEW, lo primero que se representa en pantalla es una ventana de Panel nueva sin título. Para ver conjuntamente esta pantalla y el diagrama de bloques se escogerá el menú *Windows > > Tile Left and Right o Tile Up and Down*.

En la ventana de Panel es donde se colocan todos los controles e indicadores que van a formar parte del panel frontal del VI. El diagrama de bloques también va a estar formado por un conjunto de objetos gráficos denominados terminales y nodos, los cuales van a estar unidos entre sí por medio de conexiones.

Cada una de las ventanas está provista de una serie de menús y herramientas, que sirven para la correcta colocación y utilización de estos objetos, así como para la posterior puesta en funcionamiento del VI.

##### 2.1.2.4.1 Barras de herramientas del panel frontal y diagrama de bloques


Dentro de las herramientas que más utiliza el programador están aquellas que permiten manipular el VI, es decir, poner en funcionamiento el VI, pararlo, depurarlo, etc. Según esto, hay una barra de herramientas en cada una de las ventanas (figura 5). Estas barras contienen las opciones básicas para poder manejar el panel frontal por un lado, y el diagrama de bloques por otro. Se encuentran en la parte superior del mismo.





Figura 5: Barra de herramientas del panel frontal





Botón **Run**: al pulsar este botón el VI se pone en funcionamiento.


En caso de que se haya cometido algún error este botón toma la siguiente apariencia  denominándose ahora **List Errors** y que si es pulsado nos mostrará una lista de los errores cometidos.


 Botón **Run Continuously**: si se pulsa permite que el VI se vuelva a ejecutar cuando finaliza.

 Botón **Stop**: si se pulsa la ejecución del VI se detiene. Sólo se puede usar mientras el VI este corriendo.

 Botón **Pause**: permite detener la ejecución del VI, si vuelve a pulsarse el VI continuará desde el punto en el que se detuvo.

 **Text Settings**: permite cambiar el tipo, tamaño, color o estilo del texto del panel frontal o diagrama de bloques.

 **Align Objects**: permite alinear objetos de forma vertical, horizontal, etc.

 **Distribute Object**: permite distribuir objetos de forma que la separación entre ellos no sea aleatoria.

La barra de herramientas del diagrama de bloques (figura 6) añade algunas opciones más que la del panel frontal.

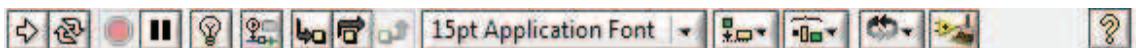






Figura 6: Barra de herramientas del diagrama de bloques

 **Highlight Execution**: si se pulsa se podrá ver como fluyen los datos en el diagrama de bloques mientras el VI se está ejecutando.

 **Step over**: permite ejecutar el VI paso a paso deteniéndose antes de ejecutar un nodo. Para continuar se ha de pulsar de nuevo este botón.

 **Step into**: funciona igual que *Step over*, lo diferencia es que si el nodo donde se detiene es un subVI, al pulsar de nuevo el botón para que continúe la ejecución paso a paso se abrirá este VI y se empezará a ejecutar el diagrama de bloques del mismo.

 **Step out**: se pulsará cuando se desee salir de un subVI, una estructura o del mismo diagrama de bloques, y finalizar así la ejecución del programa.

#### 2.1.2.4.2 Paletas

Hay tres paletas gráficas que ayudan al programador a diseñar y manejar un VI. Una de ellas, llamada paleta de herramientas o *Tools Palette* (figura 7), contiene todo lo necesario para seleccionar un objeto, moverlo, manejar un control, etc. Para visualizar la paleta se ha de seleccionar la opción *Show tools Palette* dentro del menú *Windows*. El cursor adopta la forma de la herramienta que se seleccione, de manera que para utilizarla solo se tiene que hacer *click* con el ratón sobre el objeto en el que se quiere actuar.



Figura 7: Paleta de herramientas

En las otras dos paletas se encuentran todos los elementos necesarios para construir un Instrumento Virtual. Hay una paleta disponible para cada una de las dos partes principales de un VI (panel frontal y diagrama de bloques). La asociada al panel frontal recibe el nombre de paleta de Controles (*Controls Palette*), representada en la figura 8, y contiene todos los controles e indicadores disponibles en LabVIEW. Está formada por una serie de iconos que incluyen a su vez subpaletas con distintos tipos de controles e indicadores.

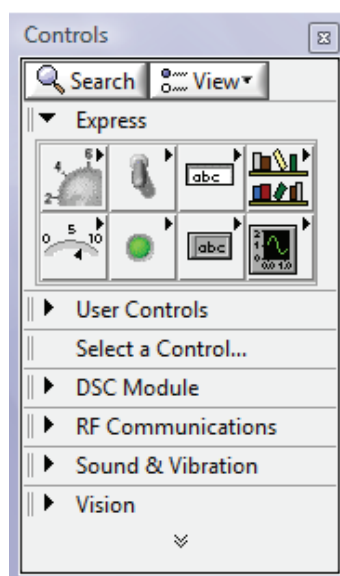


Figura 8: Paleta de controles

Para acceder a la paleta de Controles se puede seleccionar el menú *Windows >> Show Controls Palette*, o hacer *click* con el botón derecho en el panel frontal.

El diagrama de bloques se construye con ayuda de la paleta de funciones (*Functions Palette*). En ella se encuentran todos los nodos disponibles en LabVIEW. Cada una de las opciones de la paleta contiene a su vez una subpaleta con iconos de alto nivel. Si la paleta no está visible, se debe seleccionar *Show Functions Palette* dentro de menú *Windows* o hacer *click* con el botón derecho en el diagrama de bloques. En la figura 9 podemos ver la paleta de funciones.

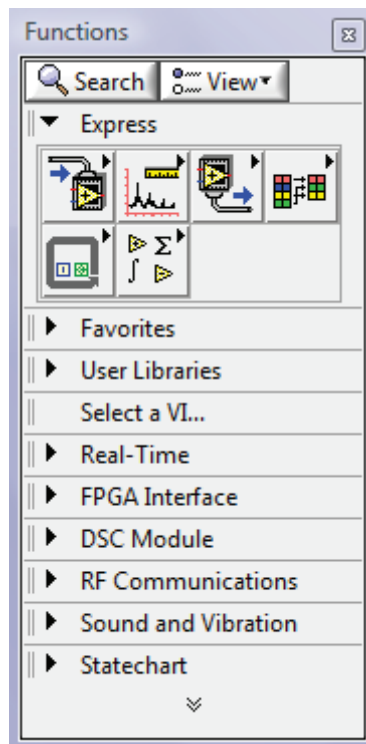


Figura 9: Paleta de funciones

Las tres clases de componentes que forman el diagrama de bloques de un VI son: nodos, terminales y cables.



#### 2.1.2.4.3 Nodos.

La función principal de un nodo es procesar los datos de su entrada, cuando estos estén disponibles, entregándolos en sus terminales de salida. En un lenguaje de programación convencional se conocen por funciones, sentencias o subrutinas. Hay cuatro tipos de nodos: funciones, subVIs, estructuras, y CINs (*Code Interface Nodes*).

- Las funciones son nodos que efectúan operaciones elementales tales como suma o resta de números, entrada/salida de archivos o manejo de *strings*.

- Los subVIs son VIs que se puede ejecutar dentro del diagrama de bloques de otros VIs.

- Las estructuras son representaciones gráficas de bucles o sentencias *Case*, por ejemplo, que se encargan de repetir bloques de código o de ejecutarlo según se dé o no una condición.

- Los CINs son interfaces entre el diagrama de bloques y un programa en C realizado por el usuario.

Todos estos componentes se encuentran dentro de la paleta de funciones. En general se representan como iconos cuadrados y muchos de ellos pueden modificar su forma y tamaño para adecuarse al número de entradas y salidas que se necesitan.

#### 2.1.2.4.4 Terminales

Los terminales son aquellos elementos a los que se dirigen los datos, “sumideros”, o desde los que salen los datos, “fuentes”. Según esto todos los nodos tienen por lo menos una fuente y un sumidero.

Otro tipo de terminales se generan cada vez que se coloca un control o indicador sobre el panel frontal. En esta ocasión, cada vez que se crea un control aparece en el diagrama de bloques un terminal asociado, que se comporta como una fuente de la que salen los datos que introduce el usuario con el control. De la misma manera, los valores obtenidos en el diagrama llegan finalmente a un terminal asociado que se comporta como un sumidero. De esta manera los datos que recibe el terminal-indicador son los que van a mostrar el programa al usuario.

#### 2.1.2.4.5 Cables.

Los cables son los elementos de unión entre los diferentes nodos y terminales. Dependiendo del tipo de terminal o nodo al que esté conectado, transportará un tipo de datos diferente. Se puede identificar rápidamente el tipo atendiendo al color y grosor del cable. Estos grosores y colores van a asociados al tipo de dato (doble, entero 32 *bits*, entero 8 *bits*, *booleano*, *array de booleanos*, etc.).

#### 2.1.2.4.6 Flujo de datos

Los programas de LabVIEW se ejecutan de acuerdo a los siguientes criterios:

- Un nodo solo se ejecuta cuando los datos están disponibles en todos sus terminales de entrada.
- El nodo proporciona los datos a los terminales de salida cuando ha terminado de ejecutarse.
- Los datos pasan de los terminales fuente a los terminales sumidero instantáneamente a través de los cables.

### 2.1.3 Desarrollo de aplicaciones de adquisición de datos con LabVIEW

#### 2.1.3.1 Introducción.

LabVIEW incluye un conjunto de VIs específicos, a los que también podemos denominar *drivers*, que permiten configurar, adquirir y enviar datos a las tarjetas de adquisición.

Los VIs utilizados son los *DAQmx* que están localizados en la paleta de funciones *DAQmx-Data Acquisition*. Como se puede ver en la figura 10, la paleta de adquisición de datos *DAQmx* (*DAQmx-Data Acquisition*) contiene todos los VIs necesarios para realizar operaciones de E/S analógica o digital y temporización. La primera ventaja es que se utilizan los mismos VIs para programar todas las operaciones de adquisición, simplificando el proceso de aprendizaje y utilización del *driver*. Además nos permite tener un mayor control sobre el proceso de medida. Estos VIs permiten programar una tarea de adquisición paso a paso, pudiendo configurar todos los parámetros necesarios. La mayoría de estos VIs son polimórficos, es decir, se pueden utilizar para realizar varias operaciones.

Cuando se trabaja con E/S analógicas las tareas pueden ser programadas para adquirir una muestra,  $n$  muestras o adquirir continuamente.

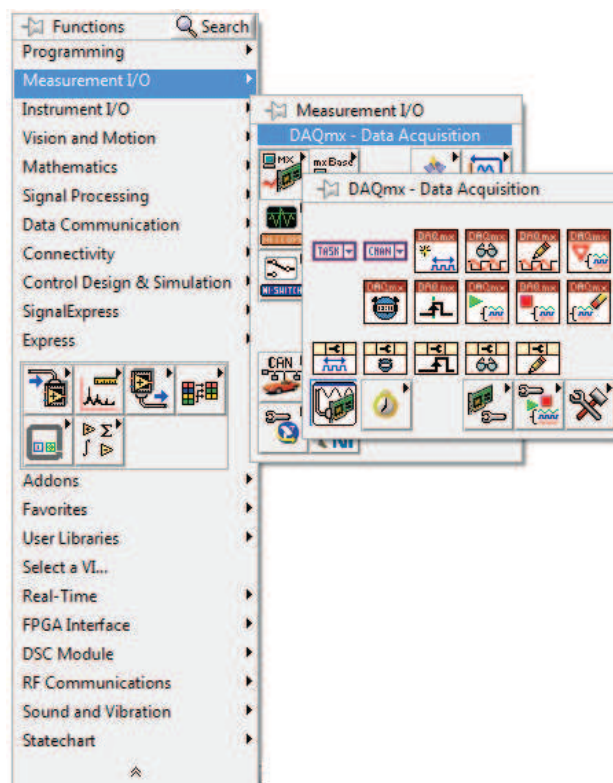


Figura 10: Paleta de adquisición de datos *DAQmx*

### *2.1.3.2 Comunicación serie*

Hay muchos instrumentos como voltímetros, generadores de señales, etc. que se pueden conectar a un ordenador por medio de un bus de comunicaciones (GPIB, RS232, etc.) pudiendo así controlar el instrumento desde un programa y automatizar su funcionamiento.

#### *2.1.3.2.1 Instrumentos serie*

Desde su aparición los ordenadores personales han incluido puertos serie para conectar dispositivos externos. El protocolo más popular ha sido el RS-232, aunque hoy en día está siendo desplazado por el bus serie universal (USB).

El estándar RS-232 es el protocolo serie más extendido para el control de instrumentos programables y en muchos sectores de la instrumentación es el interfaz de comunicaciones más utilizado.

En una comunicación serie el transmisor envía los datos bit a bit a través del canal de comunicación. Aunque es más lento que una comunicación paralelo, en la que se envía vario bits simultáneamente, es más sencillo y permite alcanzar mayores distancias.

En la transmisión se utilizan como mínimo tres líneas: transmisión (Tx), recepción (Rx) y masa (GND). Existen otras líneas adicionales que no son obligatorias. La comunicación es asíncrona y está definida por los siguientes parámetros:

- Velocidad de transmisión. Indica la velocidad de transmisión en bits por segundo o baudios.
- Bits de datos: indica el número de bits de datos que se envían en cada transferencia.
- Paridad: permite implementar un mecanismo de detección de errores sencillo. Los dos tipos de paridad más utilizados son paridad par o impar. La paridad par o impar consiste en poner el bit de paridad al valor necesario para que la transmisión contenga un número par o impar de unos. El receptor utiliza la información del bit de paridad para determinar si ha habido error en la comunicación.
- Bit de comienzo: sirve para avisar al receptor de la llegada de un nuevo paquete.
- Bits de parada: se utilizan para indicar el final de un paquete. Además permite que los dispositivos involucrados en la comunicación se sincronicen ya que al utilizar cada uno su propio reloj para enviar datos estos pueden desfasarse.

### 2.1.3.3 Driver de un instrumento

El driver de un instrumento es un conjunto de rutinas software que permiten controlar un instrumento programable (configuración, lectura, escritura...).

El cuerpo del driver de un instrumento consta de dos categorías de VIs. La primera es una colección de VIs de bajo nivel denominados *Components VIs*, son módulos individuales donde cada uno controla una funcionalidad del instrumento. La segunda comprende VIs de alto nivel llamados *Applications VIs*, que combinan VIs de bajo nivel para realizar operaciones y medidas con el instrumento. Esta categoría incluye un VI denominado *Getting Started* que se emplea para comunicarse con el instrumento sin la necesidad de cablear ningún subVI en el diagrama de bloques. Es un panel frontal virtual que permite comprobar las comunicaciones con el instrumento. Este VI suele contar de tres subVIs: un *Initialize VI*, un *Application VI* y un *Close VI*.

Los VIs de bajo nivel con los que se construyen los *Applications VIs* se dividen en seis subcategorías:

- Initialize VI*: éste debe ser el primer VI que se invoca y que permite establecer la comunicación con el instrumento.
- Configuration VIs*: permiten configurar el instrumento para realizar una operación
- Action&Status VIs*: consta de dos tipos de VIs:
  - Action VIs*: permiten que el instrumento inicie o finalice una operación
  - Status VIs*: obtienen el estado actual del instrumento o el estado de una operación
- Data VIs*: se encargan de la transferencia de datos desde el instrumento o hacia él.
- Utility VIs*: realizan operaciones que son auxiliares.
- Close VI*: todo driver debe incluir un *Close VI*, que termine la conexión software con el instrumento y libere todos los recursos.

A la hora de realizar una aplicación con un driver se colocan los VIs necesarios para cada una de las operaciones a realizar, empezando siempre por el de inicialización y acabando con los de control de errores y cierre

#### 2.1.3.3.1 Programación a bajo nivel VISA

Si no se dispone del driver de un instrumento se debe implementar la comunicación con el instrumento a bajo nivel mediante el envío de los comandos propios del instrumento y la posterior recepción de las respuestas, es decir, se han de implementar cada una de las funciones que haría el driver mediante alguna librería de comunicación.

La librería *VISA* (*Virtual Instrument Software Architecture*) ofrece un conjunto de VIs, como se puede ver en la figura 11, que permiten enviar y recibir mensajes a un instrumento programable.

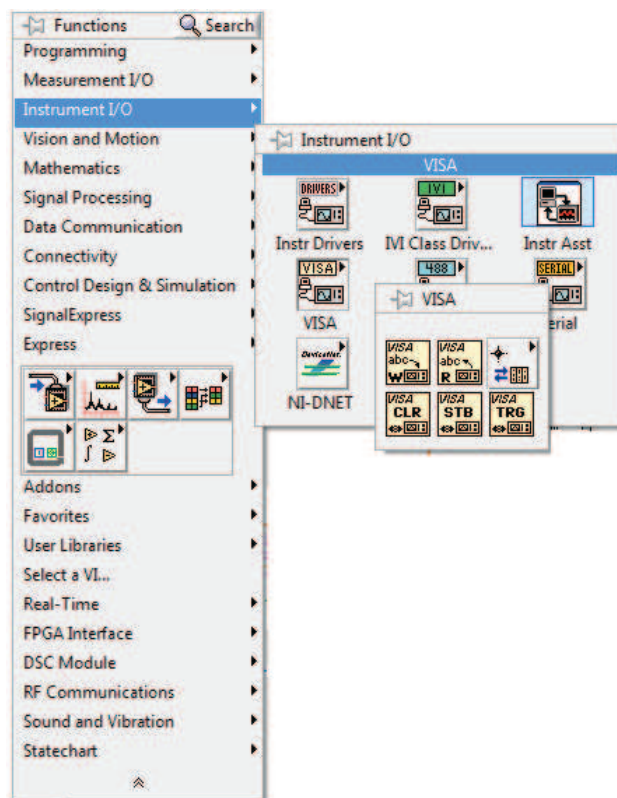


Figura 11: Paleta de adquisición de datos VISA

Las funciones *VISA Write* y *VISA Read* permiten realizar las operaciones básicas de comunicación con instrumentos ya que permiten enviar y escribir mensajes. Además de estas funciones la librería *VISA* ofrece otras funciones para realizar operaciones específicas como disparo, inicialización, sondeos, etc.

## **3. DESARROLLO DEL SISTEMA**

### 3.1 Adquisición de datos de sensores y control de una válvula

En esta parte de la memoria se va a describir uno de los dos objetivos del proyecto. El primer objetivo consiste en realizar una aplicación que permita adquirir los datos provenientes de unos sensores de presión, caudal y nivel para el estudio de sistemas hidráulicos como canales o presas. Esta aplicación ha de ser capaz de adquirir los datos de un número de sensores que puede variar en función del experimento que se vaya a realizar

En algunas ocasiones será necesario la utilización de una válvula para controlar la cantidad de agua que fluye en un experimento, para ello se tendrá que incorporar a la aplicación el código de una aplicación ya realizada para el control de la misma.

#### 3.1.1 Material utilizado y conexiones

Los tipos de sensores a utilizar son sensores de presión, sensores de nivel y sensores de caudal. Estos sensores envían señales de corriente comprendidas entre 4 y 20mA en las cuales está contenida la información del valor de la medida. Para poder recoger esta información se dispone de 4 tarjetas de *National Instruments* modelo 9208 (figura 12) a las cuales conectaremos los sensores. Estas tarjetas son del tipo *Analog Input* (AI), es decir, pueden recibir señales analógicas y, para este modelo en particular, señales de corriente.

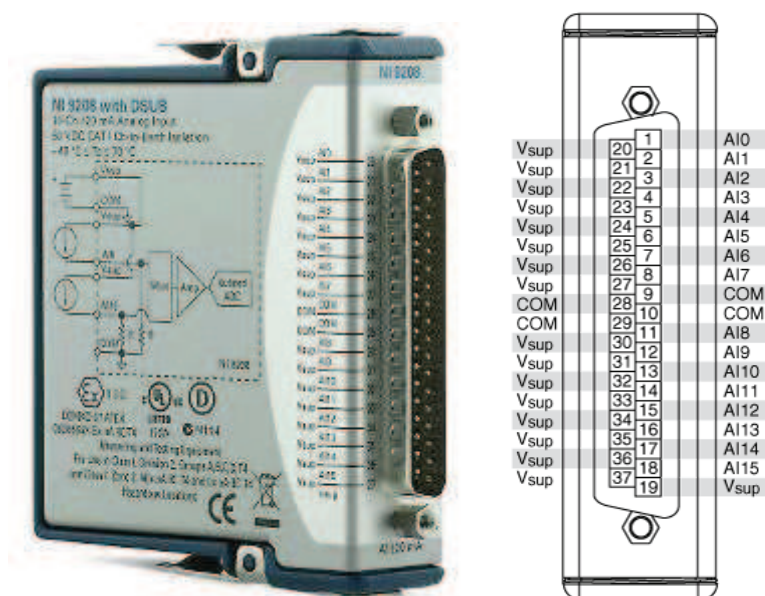
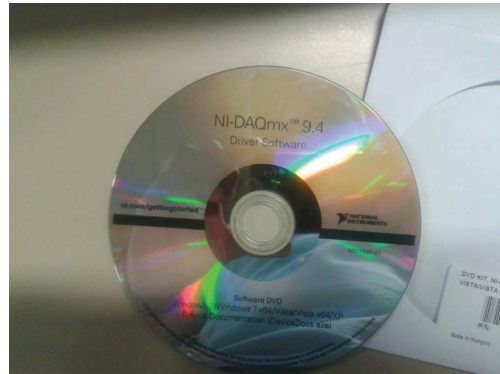


Figura 12: Tarjeta NI-9208



Estas tarjetas NI-9208 se conectarán a un chasis, *CompactDAQ* de *National Instruments* modelo 9174, con 4 ranuras a las cuales se pueden conectar más de 50 modelos diferentes de tarjetas entrada/salida entre las cuales se encuentra la tarjeta NI-9208. Este chasis tiene una conexión de alimentación a la cual se conectará la fuente que lo acompaña y una salida con interfaz USB 2.0 mediante la cual se obtendrán las medidas de los sensores conectados a las diferentes tarjetas en el ordenador. Respecto a su funcionamiento también habrá que instalar en el ordenador, antes de usar el chasis, los drivers contenidos en CD que lo acompañan. En la figura 13 se puede ver el chasis con el CD que lo acompaña.



**Figura 13: NI cDAQ-9174 y CD de drivers**

Los modelos de los sensores que se han utilizado hasta el momento son:

- Sensores de presión: XA-300 y XA-700.
- Sensores de nivel: LU05.
- Sensores de caudal: ninguno hasta el momento.

En el apartado 5 de la memoria se pueden ver las características de los sensores utilizados, de presión y nivel.

Estos sensores necesitan ser alimentados para funcionar, para ello, se dispone de una fuente de alimentación de *National Instrument* modelo PS-15 como se puede ver en la figura 14.

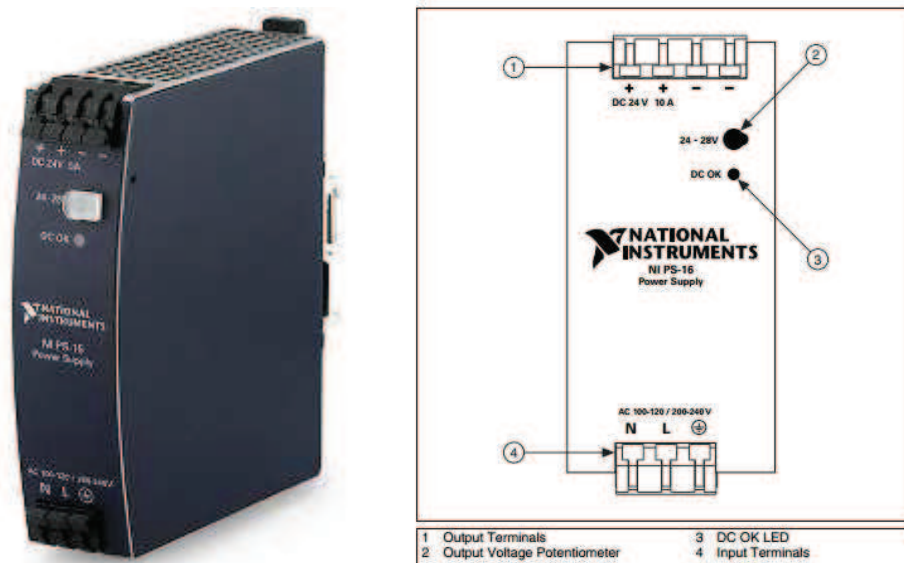


Figura 14: Fuente de alimentación PS-15

La conexión de la fuente de alimentación con los sensores se realiza a través de las tarjetas NI-9208 a la cual están conectados. Para ello hay que seguir el siguiente el esquema representado en la figura 15:

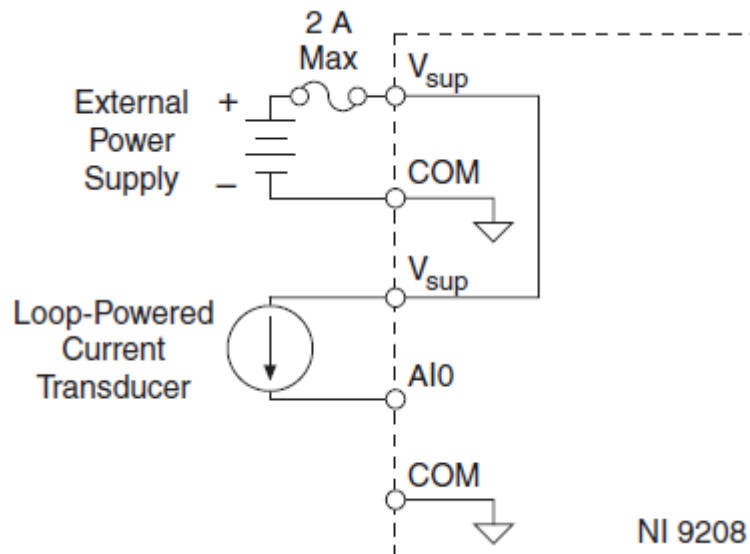
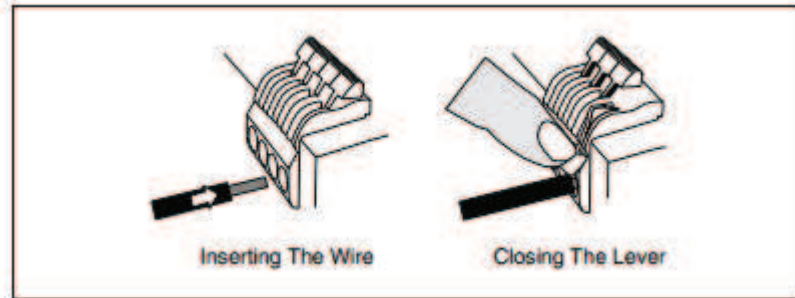


Figura 15: Conexión de la fuente de alimentación y sensores con la tarjeta NI-9208

Como se puede ver en la figura 15 conectaremos nuestra fuente de alimentación PS-15 a la tarjeta NI 9208.

De la fuente de alimentación se sacarán dos cables para alimentar los sensores, uno del positivo y otro del negativo, el positivo irá a cualquiera de los Vsup de la tarjeta y el negativo irá a un COM de la tarjeta. Como se ha dicho en líneas anteriores sirve cualquier Vsup ya que todos los Vsup de la tarjeta están unidos entre sí internamente.

Para sacar los cables tanto de las entradas como de las salidas de la fuente de alimentación hay que seguir el siguiente esquema representado en la figura16:



**Figura 16: Esquema de conexionado fuente de alimentación**

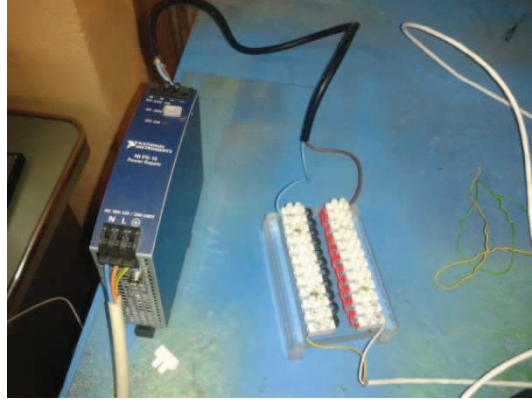


**Figura 17: Conexionado realizado con fuente de alimentación PS-15**

La imagen de la izquierda de la figura 17 se corresponde con la conexión de la red eléctrica a la fuente y la de la derecha son los cables que se han sacado de la misma.

El fabricante nos recomienda que solo se saque un cable positivo y otro negativo de la fuente de alimentación y a continuación, si es necesario, utilizar algún sistema para alimentar varios dispositivos y así evitar que se acumulen los cables en los terminales de salida.

Para esto último se han usado un conjunto de clemas, fabricadas por los técnicos del laboratorio de hidráulica, como podemos ver en la figura 18:

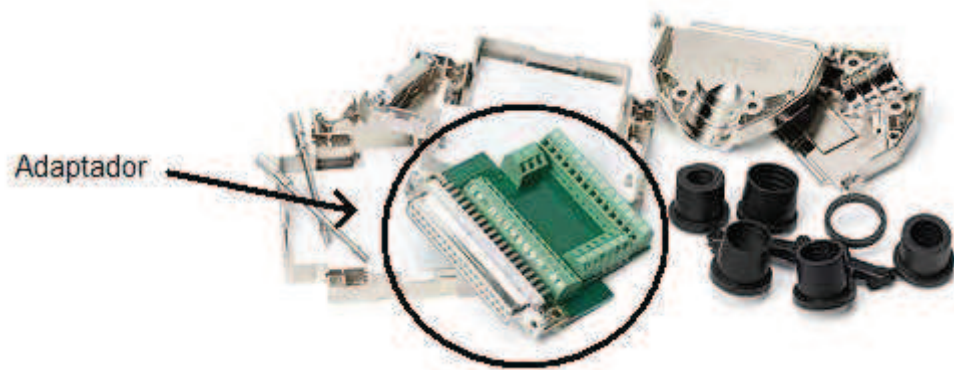


**Figura 18: Clemas**

Este conjunto de clemas permitirán obtener por separados los dos cables necesarios, positivo y negativo, que llegarán a las tarjetas NI-9208.

Respecto a los sensores se conectarán dos cables por sensor a la tarjeta, uno que va a  $V_{sup}$  y otro que va al canal AIX al que se vaya a conectar, siendo X cualquier valor comprendido entre 0 y 15. Para ver que cables del sensor son los que van al  $V_{sup}$  y al AIX se ha de consultar el manual del sensor correspondiente.

Para facilitar las conexiones a cualquiera de los terminales de las tarjetas NI-9208 se ha conectado a la entrada de la tarjeta el siguiente adaptador que se puede ver en la figura 19.



**Figura 19: Adaptador pines tarjeta NI-9208**

Para realizar cualquier conexión con la tarjeta solo será necesario introducir el cable en el agujero correspondiente y apretar el tornillo para que éste quede bien sujeto. Se aconseja que el cobre del cable que se vaya a introducir en el agujero este bien retorcido o usar el objeto de la figura 20:



**Figura 20: Objeto para el conexionado**

Si se usa el objeto de la figura 20 se introducirá el cable pelado por la parte más ancha y a continuación se ha de introducir la parte metálica, parte fina, en el agujero correspondiente y atornillar, consiguiendo así que no sobresalga ningún cable de cobre que pueda hacer conexión con cables adyacentes ya que queda protegido por la parte amarilla que es de plástico, no conductora.

### 3.1.2 Distribuciones de canales por tarjeta

A la hora de conectar un sensor a la tarjeta tenemos que tener precaución de a qué canal se está conectando ya que no se puede conectar cualquier sensor al canal que se quiera. En la tabla 1 se puede ver la distribución de canales que se ha realizado por tarjeta, es decir, si se usan 1, 2, 3 o 4 tarjetas en cada una de ellas se ha de respetar esta distribución.

Canal	Tipo de sensor
0	Nivel
1	Nivel
2	Nivel
3	Presión
4	Presión
5	Presión
6	Presión
7	Presión
8	Presión
9	Presión
10	Presión
11	Presión
12	Presión
13	Presión
14	Caudal
15	Caudal

**Tabla 1: Distribución por tarjeta para versión sin válvula**

Se han realizado dos aplicaciones, una en la que se puede controlar una válvula y otra en la que no. Si se va a utilizar la aplicación que no permite el control de la válvula todas las tarjetas que se usen deberán seguir la distribución explicada en la tabla 1, si en cambio se va a utilizar la aplicación en la que se puede usar la válvula, las conexiones de la primera tarjeta se verán modificadas y las del resto se mantendrán como la de la tabla 1.

Antes de exponer la distribución de las tarjetas para la aplicación de la válvula hay que decir que el cableado necesario para la válvula no se puede conectar a una tarjeta *Analog Input* como son las tarjetas NI-9208. Para controlar una válvula se necesita una tarjeta que

permita enviar señales para así poder darla una orden, es decir, se necesitan tarjetas *Analog Output*. En el laboratorio de hidráulica la válvula ya está conectada a una tarjeta *Analog Output* modelo NI-9264 por lo que solo será necesario colocar esa tarjeta en la cuarta ranura de nuestro chasis.

Después de lo explicado anteriormente hay que indicar que solo se podrán utilizar como máximo tres tarjetas *Analog Input* para conectar sensores ya que la última será la tarjeta *Analog Output* donde se conectará la válvula y en ésta no se pueden conectar nuestros sensores.

En la tabla 2 se puede ver la distribución de las tarjetas si se usa la aplicación en la que se puede controlar la válvula.

TARJETA 1(NI9208)		TARJETAS 2,3(NI9208)		TARJETA VÁLVULA(NI9264)	
Canal	Tipo de sensor	Canal	Tipo de sensor	Canal	Dispositivo
0	Nivel	0	Nivel	0	Válvula
1	Nivel	1	Nivel		
2	Nivel	2	Nivel		
3	Presión	3	Presión		
4	Presión	4	Presión		
5	Presión	5	Presión		
6	Presión	6	Presión		
7	Presión	7	Presión		
8	Presión	8	Presión		
9	Presión	9	Presión		
10	Presión	10	Presión		
11	Presión	11	Presión		
12	Presión	12	Presión		
13	Válvula	13	Presión		
14	Caudal	14	Caudal		
15	Caudal	15	Caudal		

Tabla 2: Distribución por tarjeta para versión con válvula



Los cambios introducidos en la distribución de la tarjeta número 1 son los dos siguientes:

Primeramente, como se verá explicado en páginas posteriores se puede controlar la válvula indicándole el caudal en litros por segundo que se desee, esto es gracias a que se ha conectado un sensor de caudal para medir el caudal que está saliendo de la válvula. Este sensor ha de conectarse obligatoriamente al canal AI14 de la tarjeta 1.

En segundo lugar, la válvula tiene una salida en corriente que indica su posición, esta salida se conectará al canal 13 de la tarjeta 1 y su conexión es prácticamente igual a la de los sensores, un cable irá al canal 13 (AI13) y el otro a un Vsup.

Como se ha explicado los cambios solo conciernen a los canales AI14 y AI13 de la tarjeta 1. Estos cambios se han introducido ya que la información que llegará del sensor de caudal y de la válvula son fundamentales para el control de la válvula.

Si una vez realizadas las conexiones en las tarjetas, éstas no van a ser modificadas, es aconsejable usar la carcasa plateada que acompaña al adaptador de la figura 19 para así proteger todas las conexiones del exterior. En la figura 21 se puede ver la carcasa que se pone para proteger las conexiones realizadas en el adaptador.



**Figura 21: Carcasa**



### 3.1.3 Base de datos: Documento de Excel

Para hacer la aplicación flexible de cara a diferentes experimentos cuyo número de sensores pueda variar se ha decidido usar un documento de Excel en el cual se puedan introducir las características de los sensores que se vayan a utilizar en el experimento. Los parámetros a introducir por sensor son los siguientes:

-Tipo: sirve para indicar el tipo de sensor. Los tipos de sensor podrán ser: presión, caudal o nivel, escritos como anteriormente, sin tildes y en minúsculas.

-Modelo: con este parámetro se indicará el modelo del sensor. Por ejemplo: XA-300, XA-700, etc. Es un parámetro identificativo para el usuario para que le sea más fácil identificar un sensor. Realmente este parámetro no se utiliza en la aplicación por lo que si no se rellena no sucederá nada.

-4mA: correspondencia con la magnitud física que mide el sensor con 4mA. Por ejemplo, un sensor de presión entrega 4mA cuando se le somete a una presión de -150mB. Las unidades a utilizar para introducir este valor serán de milibares para sensores de presión, centímetros para sensores de nivel y litros por segundo para sensores de caudal.

-20mA: correspondencia con la magnitud física que mide el sensor con 20mA. Por ejemplo, un sensor de presión entrega 20mA cuando se le somete a una presión de 300mB. Las unidades a utilizar para introducir este valor serán de milibares para sensores de presión, centímetros para sensores de nivel y litros por segundo para sensores de caudal.

-Tarjeta: número de la tarjeta a la que está conectado el sensor: 1, 2, 3 o 4.

-Puerto: número del puerto/canal al que está conectado el sensor: 0, 1, 2, 3,..., 15.

-Referencia: se utiliza para hacer un ajuste al valor que llega del sensor. Para los sensores de nivel la referencia será la altura a la que está colocado (el valor final obtenido será la referencia menos el valor que nos llega del sensor) y para los de presión nos servirá para fijar un cero (referencia) o realizar una pequeña corrección (el valor final obtenido será el valor que nos llega del sensor sumado a la referencia).

En la figura 22 podemos ver un ejemplo de un documento de Excel rellenado:

	A	B	C	D	E	F	G
1	Tipo	Modelo	4mA	20mA	Tarjeta	Puerto(AIX)	REFERENCIA
2	nivel	LU05	120	5	1	0	50
3	nivel	LU05	120	5	1	1	100
4	nivel	LU05	120	5	1	2	25
5	presion	sonda1	-150	200	1	3	0
6	presion	sonda2	-150	200	1	4	0
7	presion	sonda3	-150	200	1	5	0

Figura 22: Documento de Excel

Como se puede ver la primera fila se ha utilizado para indicar qué característica hay que poner en la columna correspondiente. La primera fila se puede utilizar para este uso o dejarse en blanco. El orden de los parámetros ha de respetarse, es decir, en la primera columna habrá que poner el tipo de sensor, a continuación el modelo y así sucesivamente hasta poner la referencia.

### 3.1.4 Aplicaciones

En esta parte nos vamos a centrar en describir los diagramas de bloques de las dos aplicaciones realizadas.

#### 3.1.4.1 Aplicación para adquirir datos de sensores

La aplicación se encuentra en su conjunto dentro de una estructura *Flat Sequence* (figura 23).

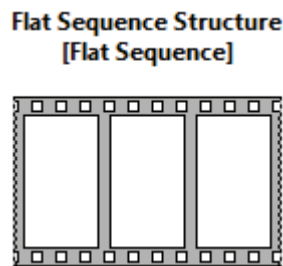


Figura 23: Estructura *Flat Sequence*

Nuestra estructura *Flat Sequence* está formada por dos subdiagramas que se ejecutan secuencialmente, primero una parte y a continuación otra asegurándonos así que el subdiagrama de la izquierda se ejecuta antes que el de la derecha.

En el primer subdiagrama, el de la izquierda, se encuentra una estructura *Stacked Sequence* (figura 24) que es igual que la estructura *Flat Sequence* pero en vez de que los subdiagramas estén uno a continuación del otro se encuentran todos juntos pudiéndose seleccionar que subdiagrama ver.

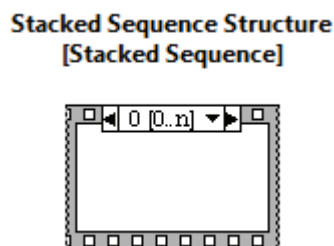


Figura 24: Estructura *Stacked Sequence*

En este primer subdiagrama de la estructura *Stacked Sequence* utilizamos la función representada en la figura 25:

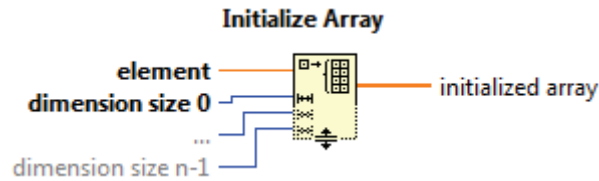


Figura 25: Función *Initialize Array*

Esta función llamada *Initialize Array* nos permite inicializar un *array* con un número de posiciones elegido a través de la entrada *dimensión size 0* y con un determinado valor introducido por la entrada *element*.

Esta función se utiliza para la inicialización, a cero, del *array* donde se guardarán los datos característicos correspondientes a cada sensor. El tamaño del *array* será el número de tarjetas (*Analog Input*) utilizadas multiplicado por 16 (número de puertos de las tarjetas) y multiplicado por el número de parámetros que guardamos por sensor, 7. El número de tarjetas se selecciona mediante un terminal asociado a un control que tenemos en el panel de control llamado *Número de tarjetas (AI)*.

En el segundo subdiagrama lo que se realiza es obtener una referencia del documento Excel donde están los datos de los sensores para poder utilizarlo en LabVIEW. Para esto utilizamos el siguiente VI llamado *NI\_report.lvclass:New Report* (figura 26).

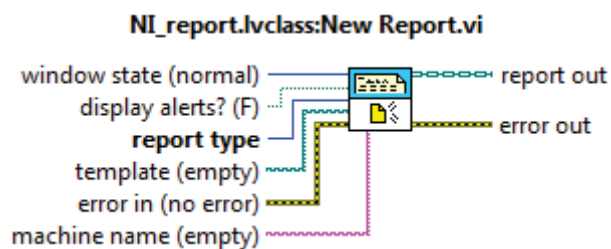


Figura 26: *New report.vi*

Los datos que se introducen de entrada son: el path donde se encuentra el documento de Excel por la entrada *template*. Para que el usuario pueda introducir ese path se ha dispuesto de un control en el panel de control cuyo terminal asociado en el diagrama de bloques se conecta a esta entrada, este control se llama *LISTA\_SENORES*. Por la entrada *report type* se indicará el tipo de documento (Excel) y por la entrada *windows state* se selecciona la opción *no change* para que no se muestre el documento en pantalla.

Por la salida *report out* se obtiene una referencia que se ha de usar cuando queramos hacer referencia al documento de Excel.

Una vez obtenida la referencia lo siguiente es copiar los datos del documento Excel a un *array* para así poder manejarlos más fácilmente. Esta parte se encuentra dentro de una estructura *for* como la de la figura 27:

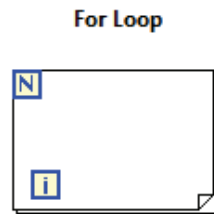


Figura 27: *For Loop*

Este bucle se ejecutará un número de veces igual al número de tarjetas utilizadas multiplicado por el número de sensores que se pueden conectar por tarjeta (16). Para indicar el número de veces que se ha de ejecutar el bucle *for* se ha conectado el resultado de la operación anterior al conector que dispone el bucle que tiene forma de "N". También se puede colocar un indicador o utilizar el valor que nos proporciona el bucle a través del cuadrado que contiene una "i" para saber el número de veces que se ha ejecutado el bucle.

Este bucle recibe la referencia del documento de Excel y el *array* inicializado a cero utilizado en la estructura *Sequence* anterior. Este *array* llega a un *shift register* o registro de desplazamiento de tal forma que la primera vez que se ejecute el bucle *for* se utilizará el *array* de ceros y las sucesivas veces se usará ese mismo *array* pero que habrá sido actualizado.

En la figura 28 se puede ver los registros de desplazamiento que son los cuadraditos con las flechas, siendo el de la flecha hacia abajo donde llega el *array* de ceros y que se usará la primera vez que se ejecuta el bucle *for*. El de la flecha hacia arriba es donde llega el *array* actualizado y que se utilizará para la siguiente iteración del bucle *for*.

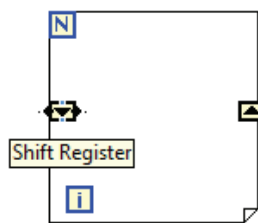


Figura 28: *Shift register* en bucle *for*

Dentro del bucle *for* lo primero que se hace es usar un VI, creado por nosotros, llamado *busqueda sensor* (figura 29), cuya misión será obtener los datos correspondientes de un sensor del documento de Excel.

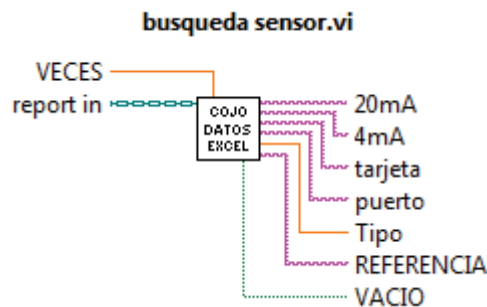


Figura 29: *busqueda sensor.vi*

Los parámetros que recibe de entrada son dos:

- La referencia al documento de Excel por la entrada *report in*.
- La fila del documento de Excel, correspondiente a un sensor del que se quiere obtener sus datos característicos, por la entrada *VECES*. A este valor se le suma uno ya que en la primera fila del Excel están los nombres que hacen referencia a que característica del sensor hay que poner en esa columna.

Los valores que nos entrega el VI son:

- 20mA: correspondencia con la magnitud física que mide el sensor con 20mA. Por ejemplo, un sensor de presión entrega 20mA cuando se le somete a una presión de 300mB.
- 4mA: correspondencia con la magnitud física que mide el sensor con 4mA. Por ejemplo, un sensor de presión entrega 4mA cuando se le somete a una presión de -150mB.
- Tarjeta: número de la tarjeta a la que está conectado el sensor: 1, 2, 3 o 4.
- Puerto: número del puerto/canal al que está conectado el sensor: 0, 1, 2, 3,..., 15.
- Tipo: nos sirve para indicar el tipo de sensor. Nos dará el valor 0 si el sensor es de presión, 1 si es de nivel o 2 si es un sensor de caudal.
- Referencia: se utiliza para hacer un ajuste al valor que nos llega del sensor. Para los sensores de nivel la referencia será la altura a la que está colocado (el valor obtenido finalmente será la referencia menos el valor que nos llega del sensor) y para los de

presión nos servirá para fijar un cero (referencia) o realizar una pequeña corrección (el valor obtenido finalmente será el valor que nos llega del sensor sumado a la referencia).

-VACIO: es una salida de tipo booleano que valdrá *true* si la primera columna del documento de Excel, que hace referencia al tipo de sensor, está vacía y *false* se contiene algún dato.

Como se explicó anteriormente lo que se realiza en este VI es obtener los datos característicos de los sensores que se encuentran en el documento de Excel, para ello se utiliza el siguiente VI dentro del VI *busqueda sensor*:

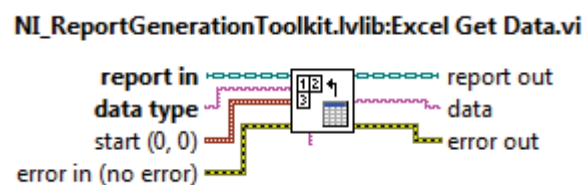


Figura 30: *Excel Get Data.vi*

El VI representado en la figura 30 permite obtener un dato específico de un documento indicándole la posición exacta y el tipo de dato del que se trata. Para ello por la entrada *report in* se indica la referencia del documento de Excel, por la entrada *data type* se indica que los datos que vamos a obtener son de tipo carácter, y por la entrada *start(0,0)* se indica la celda exacta de la que se quiere saber el valor que contiene. Para esto lo último lo que se hace es crear un *array*

Para crear este *array* lo primero que se hace es utilizar la función representada en la figura 31 llamada *Build Array*, con la que creamos un array de dos dimensiones indicando la fila y columna de la celda de la cual queremos saber el valor que contiene.

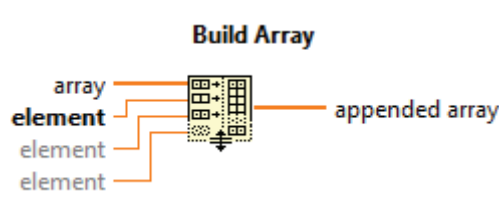


Figura 31: Función *Build Array*

A continuación se convierte ese *array* a un *Cluster* ya que el VI *Excel Get Data* requiere de este tipo de datos para indicar la posición de la celda a leer. Para ello se utiliza la siguiente función representada en la figura 32.



Figura 32: Función *Array To Cluster*

Para obtener todos los datos característicos de un sensor se han utilizado 6 VIs, *Excel Get Data*, los cuales proporcionarán por separado cada una de las características de los sensores por su salida *data*.

Cuando se obtiene el tipo de sensor no se guarda, en el caso de que el sensor sea de presión, presión, nivel o caudal, sino que se realiza una conversión, siendo ésta la siguiente:

- Si el sensor es de presión se guarda: 0.
- Si el sensor es de nivel se guarda: 1.
- Si el sensor es de caudal se guarda: 2.

Una vez listos los datos correspondientes a un sensor, proporcionados por el VI *busqueda sensor*, lo que resta a continuación es guardarlos en un *array*. Para realizar esto se utiliza el VI de la figura 33 creado por nosotros:

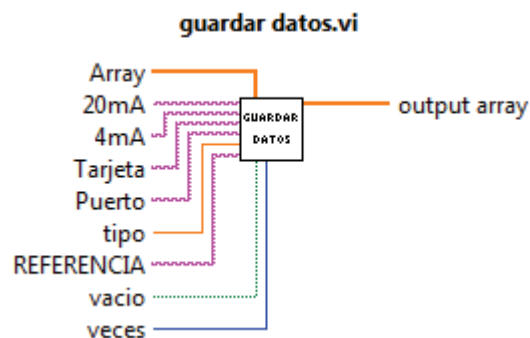


Figura 33: *guardar datos.vi*

Los datos de entrada que hay que entregarle son: el *array* donde se guardarán los datos de los sensores por la entrada *Array*, todos los parámetros característicos de los sensores por sus respectivas entradas, *20mA*, *4mA*, *tarjeta*, *puerto*, *tipo* y *referencia*, el índice del número de veces que se ha ejecutado el bucle *for* donde se encuentra este VI por



la entrada *veces* y un valor *true* o *false* por la entrada *vacío* que indique si la fila del Excel leída está vacía o no.

Todos los datos de entrada los proporciona el VI *busqueda sensor* menos dos de ellos, el *array* donde se guardarán los datos de los sensores que proviene de la estructura *Sequence* explicada en apartados anteriores y la entrada *veces* que proviene del cuadrado que contiene una “i” que se encuentra en la estructura *for* y que proporciona el número de veces que se ha ejecutado el bucle.

Para almacenar los datos característicos de los sensores en un *array* se ha optado por almacenarlos todos en una única columna almacenando los datos de un sensor a continuación de otro estando ordenados en el mismo orden que están escritos en el documento de Excel. El orden de los datos característicos de los sensores con el que se han almacenado dentro del *array* se puede ver en la tabla 3:

Posición	Característica
1ª	20mA
2ª	4mA
3ª	Pendiente
4ª	Puerto
5ª	Tipo
6ª	Referencia
7ª	Tarjeta

**Tabla 3: Orden de los datos característicos de un sensor en el *array***

Como se puede apreciar se ha almacenado un parámetro más en tercera posición del que no se había hablado anteriormente. Los sensores que se van a utilizar son lineales dentro de los límites para los que han sido calibrados. El parámetro “pendiente” se utiliza para poder realizar la conversión de la corriente proporcionada por el sensor a la magnitud física que el sensor está midiendo (presión, nivel, caudal...), por lo tanto, es necesario saber la pendiente de la recta que une el límite inferior con el límite superior que mide un sensor. La pendiente de una recta es definida como el cambio o diferencia en el eje Y dividida por el respectivo cambio en el eje X, entre 2 puntos de la recta.

Con la definición anterior el cálculo de esa pendiente es sencillo ya que en el eje Y tenemos la magnitud física que mide el sensor y en el eje X tenemos miliamperios que van desde 4 hasta 20:

$$m = \frac{20mA - 4mA}{16}$$

Donde:

-m: es la pendiente de la recta.

-20mA: es la correspondencia con la magnitud física que mide el sensor, por ejemplo, un sensor de presión entrega 20mA cuando se le somete a una presión de 300mB.

-4mA: es la correspondencia con la magnitud física que mide el sensor, por ejemplo, un sensor de presión entrega 4mA cuando se le somete a una presión de -150mB.

-16: se obtiene de restar 20mA-4mA (cambio en el eje X).

El valor en unidades de la pendiente dependerá del tipo de sensor, si el sensor es de presión sus unidades serán: mB/mA, si es de nivel será de cm/mA y si es de caudal (l/s)/mA.

Volviendo a la descripción de la función de este VI, todo lo contenido dentro del mismo se encuentra dentro de una estructura *Case* (figura 34) que permite ejecutar una parte del código u otra (*true* o *false*) en función del valor que tengamos conectado en el selector (símbolo de interrogación)

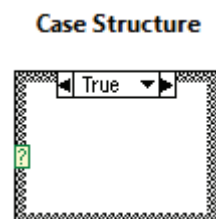


Figura 34: Estructura Case

Al selector se ha conectado el valor de la salida *vacío* que sale del VI *busqueda sensor* pasando antes por un inversor para que cuando *vacío* sea *true*, porque la celda del tipo de sensor este vacía, al VI *guardar datos* llegue el valor *false* y no se guarden los datos en el *array*, de la misma manera si la celda del tipo de sensor no está vacía al VI *guardar datos* llegará el valor *true* que hará que se guarden todos los parámetros.

Para almacenar un dato en la posición del array que se quiera se utiliza la función de la figura 35:

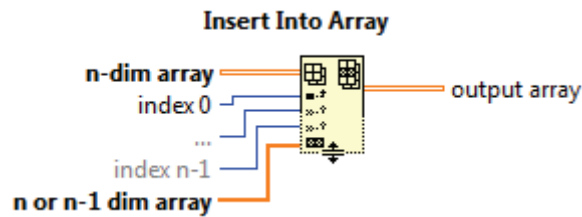


Figura 35: Función *Insert Into Array*

Los datos de entrada que hay que proporcionarle son: el *array* en el cual se quiere insertar un nuevo elemento por la entrada *n-dim array*, la posición del *array* donde se quiere insertar por la entrada *index* y el nuevo elemento a insertar por la entrada *n or n-1 dim array*.

El dato de salida obtenido será el *array* que hemos introducido en la entrada pero con un elemento nuevo en la posición indicada.

El *array* de entrada donde se guardarán los datos de los sensores proviene de la estructura *Sequence* explicada en apartados anteriores. La posición en la que se va a guardar el nuevo elemento viene determinada por el valor de *veces*, *veces* es el indicador del número de veces que se ha ejecutado el bucle *for* que como mínimo será 0 y como máximo será el número de tarjetas que vayamos a utilizar multiplicado los dieciséis sensores que se pueden conectar por tarjeta.

La primera vez que se ejecute el VI *guardar datos* *veces* valdrá cero y los datos a almacenar en el *array* serán los del primer sensor por lo que los datos de ese sensor se almacenarán en las posiciones del *array* que van desde la cero hasta la seis. Para el siguiente sensor *veces* valdrá uno y sus datos se almacenarán en las posiciones del *array* que van desde la siete hasta la trece y así sucesivamente hasta guardar todos los datos de todos los sensores.

Visto lo anterior la selección de la posición donde se quiere guardar el dato será siempre el valor de *veces* multiplicado siete (ya que son siete elementos por sensor los que se van a almacenar) y a continuación se sumará un valor para mantener el orden de posiciones descrito en la tabla 3, esos valores son los siguientes:

-Para el parámetro de 20mA no se sumará nada ya que cuando se trata del primer sensor éste parámetro irá en la posición cero, si es del segundo sensor será la posición número siete, del tercer sensor la posición será la catorce y así

sucesivamente. El valor cero, siete, catorce,... se obtiene al multiplicar veces por siete por lo que no es necesario sumarle nada.

-Para almacenar el parámetro de 4mA se sumará uno ya que está a continuación del parámetro de 20mA.

-Para el resto de parámetros que restan los valores que hay que sumar está representados en la tabla 4:

Parámetro	Suma
Pendiente	+2
Puerto	+3
Tipo	+4
Referencia	+5
Tarjeta	+6

Tabla 4: Operaciones para mantener el orden de posiciones

Respecto al dato a insertar en el *array* hay que tener en cuenta que el *array* donde se insertan los datos es un *array* de ceros, es decir, de números y algunos de los elementos que se quieren insertar son caracteres por lo que habrá que convertir los elemento que sean caracteres a números para poder insertarlos dentro del *array*. Para realizar esta conversión se utiliza la función de la figura 36:

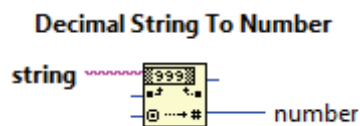


Figura 36: Función *Decimal String To Number*

Ya que todos los datos que se tienen son números, aunque estén considerados como caracteres, esta función nos sirve ya que nos devuelve el mismo valor que se tenía pero ahora el dato que devuelto es de tipo entero decimal.

Los datos que tenemos que convertir son: 20mA, 4mA, *puerto*, *referencia* y *tarjeta*.

Una vez se hayan insertados los datos característicos de un sensor este *array* llega al *shift register* para que la próxima vez que se vuelva a ejecutar el bucle *for* se vuelvan a insertar los datos característicos de otro sensor sobre el *array* estando este actualizado.

Una vez copiados todos los datos del Excel a un *array* llamado *medidas* el siguiente paso a realizar es obtener los valores que están midiendo los sensores.

Esta parte del programa se encuentra en el segundo subdiagrama de la estructura *Flat Sequence* descrita en la figura 23. Además de obtener los valores que están midiendo los sensores también se realizan las conversiones de corriente a la magnitud física correspondiente que mida un sensor, se representan en un gráfico y si se desea, o no, se guardan las mediciones en un documento de texto.

Lo primero que se realiza en este subdiagrama es crear un enlace con cada una de las tarjetas que se estén usando para a partir de este momento poder obtener los datos con una determinada frecuencia.

Para esta parte se han usado tres estructuras *case* cuyos selectores *true/false* provienen de comparar el valor introducido a través del control: *número de tarjetas (AI)* con uno, dos, tres o cuatro respectivamente. El valor de este selector puede variar entre uno y cuatro. Para realizar la comparación se usa la función de la figura 37:

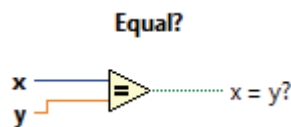


Figura 37: Función Equal?

Esta función entregará el valor *true* si “x” e “y” son iguales y *false* si son distintos.

Lo que se realiza dentro de estas estructuras *case*, si el selector es *true*, es crear un enlace con la tarjeta correspondiente e indicar el periodo de muestreo de la misma. Se utiliza una estructura *case* cuyo selector, *true* o *false*, se basa en saber si usamos una, dos, tres o cuatro tarjetas para crear enlaces a las tarjetas que se estén utilizando ya que no tiene sentido crea un enlace a una tarjeta que no están conectadas y no se vayan a utilizar.

La creación del enlace o tarea con cada una de las tarjetas se encuentran cada una por separado dentro de una estructura *case* diferente salvo una excepción. Dado que el control *número de tarjetas (AI)* tiene como valor mínimo uno siempre será necesario crear el enlace con la tarjeta conectada en la posición número uno por lo que la creación del enlace con esta tarjeta no está dentro de una estructura *case*.

El VI utilizado para crear un enlace con una tarjeta es el representado en la figura 38:

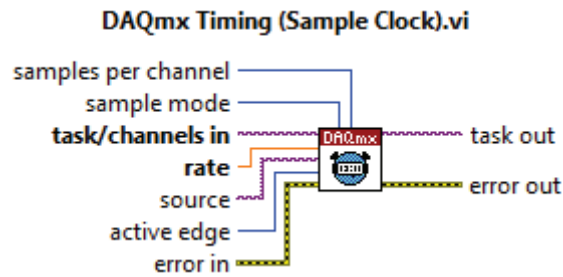


Figura 38: DAQmx Timing (Sample Clock).vi

Como parámetros de entrada se introducen: la tarjeta de la que se quieren obtener los datos a través de *task/channels in*, por la entrada *sample mode* se indica que el modo de muestreo sea continuo, la frecuencia de muestreo en muestras por canal por segundo a través de la entrada *rate* y el tamaño de un buffer, que será igual al valor de las muestras por canal, introducido a través de la entrada *samples per channel*.

Para poder asignar la tarjeta de la que se quieren obtener los datos ésta debe de estar conectada para poder así seleccionarla con la entrada *task/channels in*.

Dentro de los diferentes modos de muestreo se ha elegido la opción de muestreo continuo ya que se quieren obtener muestras hasta que se detenga la aplicación, en vez de un número de muestras concreto (muestreo finito).

Para seleccionar la frecuencia de muestreo en muestras por segundo por canal se ha creado un control en el panel de control llamado *Muestras/canal/segundo* que permita seleccionar este parámetro. Los valores de este selector oscilan entre uno y treinta ya que, como indican las especificaciones de las tarjetas que se están usando, se pueden llegar a obtener 500 muestras por segundo de forma no simultanea, es decir, que la suma de las muestras obtenidas por todos los canales puede ser de hasta quinientas en un segundo, de esta forma, dividiendo este número entre los dieciséis canales que tiene la tarjeta podemos llegar a obtener hasta 31,25 muestras por segundo por canal.

Se ha utilizado un único selector, *Muestras/canal/segundo*, para todas las tarjetas de tal forma que todas se muestrearán a la misma frecuencia para así evitar confusiones entre las tarjetas.

El tamaño del *buffer* que se selecciona a través de la entrada *samples per channel* es de mil ya que, según indica LabVIEW a través de la tabla 5, es lo recomendado si adquirimos entre cero y cien muestras por segundo por canal.

Sample Rate	Buffer Size
no rate specified	10 kS
0–100 S/s	1 kS
100–10,000 S/s	10 kS
10,000–1,000,000 S/s	100 kS
>1,000,000 S/s	1 MS

Tabla 5: Tamaño del *buffer*

Las salidas de este VI son dos: *task out* o tarea de salida en la que van incluidas todas las características introducidas a través de los parámetros de entrada, y la salida *error out* que contiene información interesante si se produjera algún error.

Una vez introducida esta información lo que procede a continuación es comenzar a obtener muestras del número de tarjetas que se haya elegido. Lo que resta del programa y que incluye lo comentado en la línea anterior se encuentra dentro de un bucle *while* como el de la figura 39.

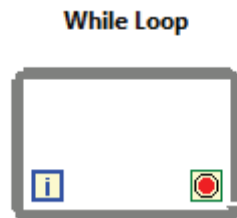


Figura 39: Bucle *while*

Un bucle *while* se ejecuta un número de veces determinado porque se cumpla o no, *true* o *false*, una condición. Esa condición habrá que conectarla al punto rojo que vemos en la parte inferior izquierda de la figura 39. Lo otro que se puede ver en la figura 39 es un cuadradito con una “i” dentro que nos proporciona el número de veces que se ha ejecutado el bucle *while*.

Al punto rojo hemos conectado un control llamado *terminar* que nos permitirá salir de este bucle *while* parando la aplicación en el momento que se pulse. Se trata de un pulsador que cuando sea pulsado entregará el valor *true* parando así la aplicación, si no se pulsa entrega el valor *false* y la aplicación continuará ejecutándose.

Lo primero que se realiza dentro de este bucle *while* es la obtención de las muestras de las tarjetas que se estén usando, para ello se utiliza el VI de la figura 40:

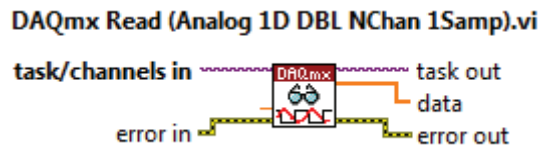


Figura 40: *DAQmx Read (Analog 1D DBL NChan 1Samp).vi*

Este VI nos permite leer u obtener una muestra según las características introducidas a través de la entrada *task/channels in*. En este caso se obtendrán dieciséis muestras dentro de un *array* de una dimensión por la salida *data*, se obtiene un *array* de dieciséis posiciones ya que se ha seleccionado previamente una tarjeta que está formada por dieciséis canales, si se hubiera elegido solo un canal se obtendría un único dato en vez de un *array*. La frecuencia con la que se obtienen los datos será la que se haya elegido mediante el control *Muestras/canal/segundo*. En la entrada *error in* conectaremos la salida *error out* del VI *DAQmx Timing* que se ha explicado anteriormente.

Las salidas obtenidas son: el *array* de datos, *task out* (referencia a la tarea antes de haberse ejecutado) y *error out* que contiene la misma información que *error in* si no se ha producido ningún error en este VI, si se produjese se añadiría automáticamente la información correspondiente al error.

En la aplicación se han emplazado cuatro VIs como este, uno por tarjeta, siguiendo el mismo razonamiento que con el VI *DAQmx Timing* situando los que hacen referencia a las tarjetas dos, tres y cuatro dentro de una estructura *case* cada uno y cuyos selectores están controlados de la misma forma que el VI *DAQmx Timing*. El VI de la tarjeta uno no está dentro de ninguna estructura *case*.

Cada uno de estos VIs *DAQmx Read* está unido al VI *DAQmx Timing* que haga referencia a la misma tarjeta mediante la unión de *task out* y *error out* de los *DAQmx Timing* con los respectivos *task in* y *error in* de los *DAQmx Read*.

Una vez se tenga listo en las salidas de los *DAQmx Read* las muestras obtenidas de leer los canales de las tarjetas correspondientes toca convertir esos valores de corriente a la magnitudes físicas que lean los sensores ya que las tarjetas leen las corrientes que nos entregan los sensores.

Para ello se ha utilizado un bucle *for* que se ejecute tantas veces como tarjetas hayamos seleccionado usar multiplicado por los dieciséis sensores que hay por tarjeta para



así ir cogiendo los datos de cada uno de los sensores, transformarlos a la magnitud física que midan y realizar los ajustes deseados usando las referencias introducidas en el documento de Excel.

Se han creado dos VIs (figura 41) para llevar a cabo todo lo anterior: *calculos.vi* y *ajustes.vi*

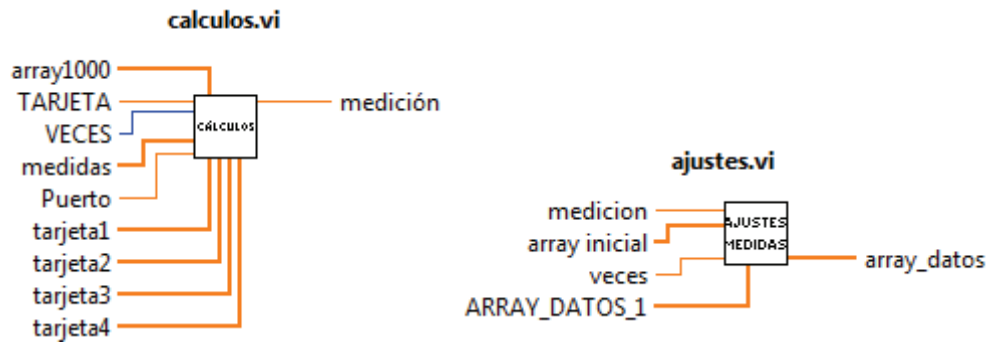


Figura 41: *calculos.vi* y *ajustes.vi*

El primero, *calculos*, servirá para hacer la transformación de corriente a la magnitud física que miden los sensores y el segundo, *ajustes*, para realizar los ajustes deseados a las medidas obtenidas usando las referencias.

Como se puede ver en la figura 41 *calculos.vi* recibe varios parámetros de entrada:

- Entrada *array1000*: por aquí recibe un *array* que tiene tantas posiciones como tarjetas hayamos seleccionado multiplicado por los 16 sensores que puede haber en cada tarjeta, y en cada una de esas posiciones tiene el valor 1000. Este *array* se crea ya que los datos que nos llegan de las tarjetas vienen en amperios y si se trabaja con ellos estamos pasando por alto cierta información y se estaría usando valores como 0,001A o 0,002A mientras que si se multiplica por 1000, pasándolo a miliamperios, se usará un valor mucho más preciso.

La creación de este *array* se realiza utilizando la función *Initialize Array* que ya se ha explicado anteriormente.

- Entrada *TARJETA*: por aquí se indicará la tarjeta a la que está conectado el sensor cuya medida va a ser convertida de corriente a la magnitud física que mida el sensor.

- Entrada *VECES*: valor numérico del número de veces que se ha ejecutado el bucle *for*.

- Entrada *medidas*: *array* donde están guardados todos los parámetros característicos de cada uno de los sensores.

-Entrada *Puerto*: valor numérico que se corresponde con el número del canal o puerto de la tarjeta a la que está conectado el sensor cuya medida va a ser convertida a otra magnitud física.

-Entradas *tarjeta1*, *tarjeta2*, *tarjeta3* y *tarjeta4*: por estas entradas llegarán los arrays con las medidas obtenidas de los *DAQmx Read* correspondientes a cada una de las tarjetas.

Para indicar la tarjeta y el puerto se utiliza la función *Index Array* (figura 42):

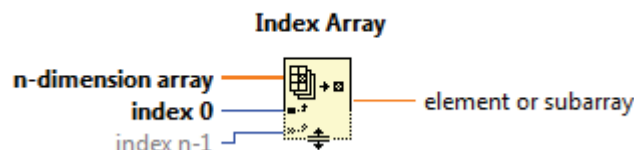


Figura 42: Función *Index Array*

Esta función devuelve el valor de la posición, indicada por la entrada *index 0*, del array introducido por la entrada *n-dimension array*, por la salida *element or subarray*.

El array introducido por la entrada *n-dimension array* es *medidas* que es donde se encuentran los datos característicos de los sensores, entre ellos la tarjeta y el puerto a la que está conectado el sensor.

Para indicar la posición del array hay que tener en cuenta lo que se explicó al describir el VI *guardar datos*, cómo están guardados los datos característicos de los sensores para así obtenerlos correctamente ahora.

La salida obtenida de este VI es la medición correspondiente a un sensor cuyas unidades serán de presión (mB), nivel (cm) o caudal (l/s) en función del tipo de sensor que sea.

Dentro de este VI lo primero que se hace es, sabiendo la tarjeta a la que está conectado el sensor introducida por la entrada *TARJETA*, quedarse con el array de medidas de la *tarjeta 1*, *2*, *3* o *4*. Para ello se utilizan estructuras *case* para quedarse con el array de medidas de la tarjeta adecuada.

A continuación se multiplica el array de medidas de la tarjeta correspondiente por mil utilizando el array introducido por la entrada *array1000*.

Una vez se tengan todas las posiciones del array en miliamperios hay que quedarse con el sensor que toque en función del índice del número de veces que se haya ejecutado el

bucle *for*. Para ello se usa la función *Index Array* y la entrada *Puerto* para seleccionar la posición del *array* donde se encuentre el sensor en cuestión.

Teniendo una medida correspondiente a un sensor toca hacer la conversión de magnitud. Los sensores que se van a utilizar son lineales entre sus extremos, 4-20mA. Recordado esto la forma más fácil de realizar la conversión es valerse de la ecuación de la recta que representa la linealidad de esos sensores.

La ecuación que hemos usado es la ecuación punto-pendiente:

$$y = m(x - x_1) + y_1.$$

Donde:

-“y” es el resultado de la conversión.

-“m” es la pendiente de la recta que se calculo en el VI guardar datos.

-“x” es la medida en miliamperios.

-“x<sub>1</sub>” vale siempre 4.

-“y<sub>1</sub>” correspondencia de la magnitud que mide el sensor con 4mA.

Descritos los parámetros de la recta se necesita obtener del *array* donde se almacenan los datos característicos de los sensores la pendiente y la correspondencia de la magnitud que mide el sensor con 4mA, para ellos se vuelve a utilizar la función *Index Array*.

Para implementar la ecuación de la recta en LabVIEW se ha utilizado lo que se conoce como *Formula Node* (figura 43):

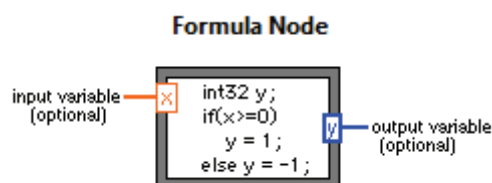


Figura 43: *Formula Node*

Un *Formula Node* permite implementar una función a través del lenguaje de programación C. Permite introducir el valor de las variables de entrada a través de unas cajitas situadas a la izquierda (*input variable*) y obtener el valor de otras variables a través de unas cajitas similares situadas a la derecha (*output variable*).

La ecuación implementada con el *Formula Node* es la misma pero se han utilizado otros nombres:

-“y”=medición.

-“inicial”= $y_1$ .

-“m”=m.

-“lectura”=x.

Para finalizar la descripción de este VI queda un detalle por explicar en relación con lo que sucede si se convierte una medida de un canal de una tarjeta en la que no hay conectado un sensor. De primeras el valor que se estaría midiendo sería 0A y a la hora de convertir ese valor darían cosas raras. Para evitar esto se ha decidido fijar un valor que represente esta situación que es frecuente que se dé ya que, por ejemplo, si se está usando una tarjeta y no todos sus canales están conectados a algún sensor se daría la situación de la que se está hablando. Este valor es el 9999, un valor que se ha elegido porque, a priori, con los sensores que se van a utilizar es un valor que ninguno de ellos va a entregar.

Otra situación curiosa que se puede dar y que puede dar lugar a error es qué sucedería si se somete a un sensor a una situación de presión, nivel o caudal que esté fuera del rango para el que ha sido calibrado. Los sensores usados, ante una situación como la descrita, entregan una corriente superior a 20mA si se los somete a una situación por encima o por debajo de los límites para los que ha sido calibrado. Ante esta situación se ha decidido optar por una solución como la del apartado anterior pero eligiendo otro valor significativo, 8888.

El otro VI del que se ha hablado, figura 41, antes es *ajustes*. En este VI lo que se hace es realizar los ajustes deseados usando las referencias introducidas a través del documento de Excel.

Los parámetros de entrada que recibe son: la salida medición del VI *calculos* a través de la entrada *medicion*, por la entrada *array inicial* recibe el *array medidas* que contiene los datos característicos de todos los sensores, el índice del número de veces que se ha ejecutado el bucle *for* a través de la entrada *veces* y por la entrada *ARRAY\_DATOS\_1* recibe un *array* inicializado a 0 de tantas posiciones como tarjetas haya multiplicado por los 16 sensores que puede haber por cada tarjeta y que será utilizado para almacenar en él las medidas de los sensores con sus correcciones, si son necesarias, que se van a realizar en este VI.

El uso que se hace del *array* que llega a la entrada *ARRAY\_DATOS\_1* de este VI es el mismo que se realizaba con el *array* en el que se guardaban los datos característicos de los sensores. Lo primero que se hace es inicializarlo a cero como se ha comentado antes usando la función *Initialize Array* y a continuación se conecta a un *shift register* (registro de desplazamiento) que está en el bucle *for* que contiene al VI *ajustes*, de esta forma la primera vez que se ejecute el bucle *for* se utilizará el *array* inicializado a cero y las sucesivas veces se usará ese mismo *array* pero que habrá sido actualizado.

Como parámetro de salida se obtienen, por la salida *array\_datos*, el *array* donde estarán guardadas las medidas de los sensores con las correcciones pertinentes. Esta salida estará conectada al *shift register* situado a la derecha del bucle *for* que contiene a este VI para que así esté actualizado para la siguiente iteración del bucle *for*.

Dentro de este VI lo primero que se hace, como se ha hecho en numerosas veces, es obtener del *array* que contiene los datos característicos de los sensores la tarjeta, el puerto y el tipo de sensor del sensor que toque.

El puerto y la tarjeta a la que está conectado el sensor se quieren para ordenar las medidas de los sensores dentro del *array* que las va a contener, de tal forma que si la tarjeta a la que está conectado el sensor es la número uno, la posición del *array* en la que se guardará la medida será la del propio puerto que irá desde la posición cero a la quince. Si la tarjeta a la que estuviera conectado el sensor fuera la número dos, tres o cuatro, la posición en la que se guardaría su medida sería la del puerto donde estuviera conectado sumado con dieciséis, treinta y dos o cuarenta y ocho respectivamente.

El otro parámetro obtenido es el tipo de sensor y la referencia. El tipo de sensor servirá para saber que tratamiento se tiene que dar a la medición que proviene del sensor. Si el sensor es de nivel se realizará la resta de la referencia, introducida para ese sensor en el documento Excel, menos la medida que entregada por el sensor. Se realiza esta operación ya que el objetivo de los sensores de nivel es medir precisamente el nivel de agua por lo que la referencia de ese sensor tendrá que ser la altura a la que esté situado, o dicho de otra forma, el nivel cero de agua. Si el sensor es de presión la operación realizada será la suma de la referencia de este sensor más la medida que nos esté entregando. En el caso de los sensores de presión la referencia tiene un objetivo similar al de los sensores de nivel que es, como su propio nombre indica, marcar una referencia. La diferencia con los sensores de nivel es que si no se introduce una referencia para los sensores de presión no sucede nada mientras que si no se introduce para los de nivel se estarían obteniendo valores negativos que pueden dar lugar a error.

Para el otro tipo de sensor que queda, caudal, no se realizará ninguna modificación con la referencia ya que no es necesaria.

Recordar también que por este VI van a pasar también las medidas de sensores que o bien no están conectados o están siendo sometidos a situaciones fuera de los límites para los que han sido calibrados. Con este tipo de valores, 8888 o 9999, no se realizará ninguna operación, se dejarán pasar sin hacerles ningún tipo de modificación con sus respectivas referencias ya que no tendría ningún sentido hacérselas.

Una vez que se tienen los valores a guardar en el *array* y sus posiciones respectivas solo falta insertarlos en el *array*. Para ellos se ha usado la función de la figura 44:

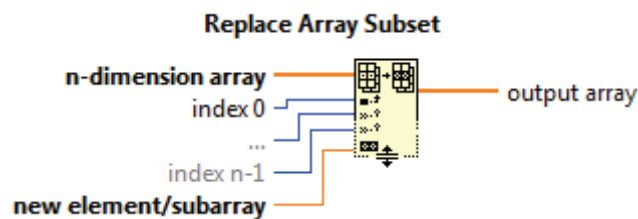


Figura 44: Función *Array Subset*

Esta función permite reemplazar el dato del *array* introducido por la entrada *n-dimensión array* cuya posición sea la indicada por la entrada *index 0*, por el dato introducido por la entrada *new element*.

Una vez terminado de explicar el bucle *for* que contiene a los VIs *calculos* y *ajustes* falta por explicar algunas partes de código que se encuentran fuera del bucle *for* pero a la vez dentro del bucle *while*.

La primera cosa que se va a explicar es la parte de los gráficos que permiten visualizar, en tiempo real, las medidas que están entregando los sensores. Para ello se han utilizado tres gráficos, uno para cada tipo de sensor: presión, nivel y caudal.

Para gestionar estos tres gráficos y proporcionarles a cada uno las medidas de los tipos de sensores que van a representar se ha creado el VI de la figura 45 llamado: *gestión\_graficos*.

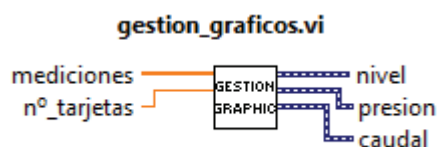


Figura 45: *gestion\_graficos.vi*

Los parámetros que recibe de entrada son: el *array* que contiene las mediciones de todos los sensores procedente del VI ajustes por la entrada *mediciones* y el número de tarjetas que se están usando por la entrada *nº\_tarjetas*.

Se obtienen tres salidas: *nivel*, *presion* y *caudal* que contienen todas las medidas de los tipos de sensores que representan respectivamente y que acabarán en su respectivo gráfico.

Dentro de este VI se obtienen los elementos del *array* que contiene las medidas de los sensores separándolos por tarjeta, es decir, si a través de la entrada *nº\_tarjetas* se indica que solo se está usando una tarjeta se obtienen las dieciséis primeras posiciones del *array*, si se usan dos tarjetas se obtendrán las treinta y dos primeras posiciones, si son tres las cuarenta y ocho primeras y si son cuatro todas las posiciones, sesenta y cuatro.

Para ello se utiliza como se ha comentado antes la función *Index Array* y estructuras *case* para no obtener posiciones del *array* correspondientes a tarjetas que no se están usando.

Una vez obtenidas las medidas necesarias en función del número de tarjetas usadas se agrupan las medidas correspondientes a un mismo tipo de sensor dentro de una tarjeta. Como se ha comentado en la explicación del documento de Excel estas son: las tres primeras para sensores de nivel, las once siguientes para sensores de presión y las dos restantes para sensores de caudal.

Para agrupar tanto las medidas de un mismo tipo de sensor dentro de una tarjeta como las de varias tarjetas se ha utilizado la función de la figura 46:

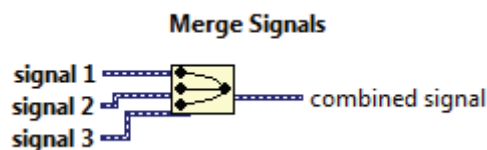


Figura 46: función *Merge Signals*

Esta función no solo permite juntar tres señales, como se ve en la figura 46, si no que se puede ampliar al número de señales que se desee teniéndolas todas juntas en la salida *combined signal*.

Una vez juntadas las medidas correspondientes a un mismo tipo de sensor dentro de una tarjeta se vuelve a utilizar la función *Merge Signals* para juntarlas con los mismos tipos de sensores de las otras tarjetas que se estén usando. De nuevo, para no juntar medidas de

sensores del mismo tipo de tarjetas que no se estén usando se vuelve a utilizar estructuras *case*.

Explicado todo esto se tendrán tres señales correspondientes a sensores de nivel, presión y caudal que irán al gráfico correspondiente cada una.

Los gráficos utilizados para representar las señales son indicadores *Waveform Chart* como los que aparecen en la figura 47 con los nombres de caudal, nivel y presión.

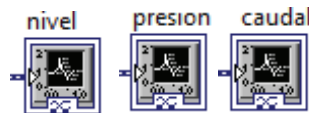


Figura 47: indicadores *Waveform Chart*

Para representar todas las señales que llegan a cada uno de ellos hay que ponerse en el peor de los casos, es decir, cuando se estén usando las cuatro tarjetas con todos sus canales ocupados. El número de sensores de nivel utilizados sería de doce, de presión cuarenta y cuatro y de caudal ocho.

Para indicarle al gráfico el número de trazas a representar se hará *click*, con el botón derecho del ratón, encima del indicador, se pincha en *properties* y a continuación saldrá un menú en el cual hay que indicar el número de trazas que se quieren visualizar. Para el indicador de nivel serán doce, para presión cuarenta y cuatro y para caudal ocho. En la figura 48 se puede ver, marcado con un recuadro, donde indicarlo.

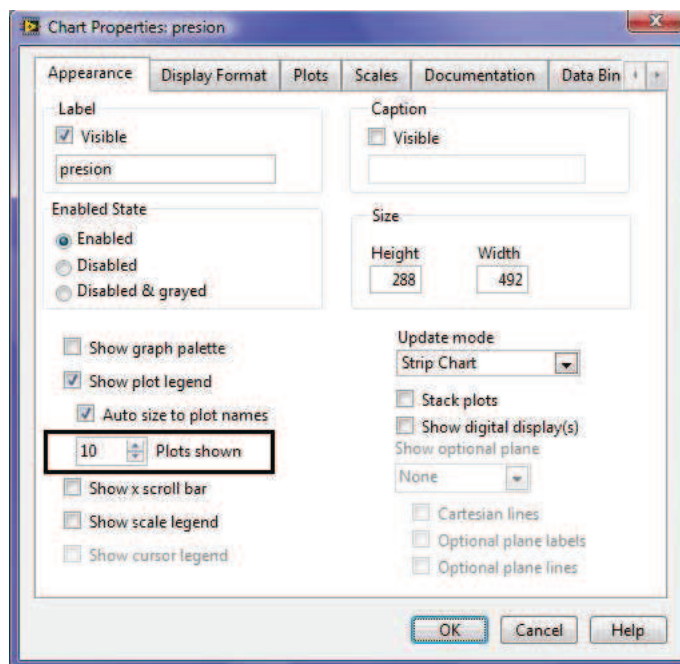


Figura 48: *Properties Waveform Chart*



Si vamos al menú *Plots* se podrá dar un nombre, tipo de línea, color, etc a cada una de las trazas.

Para el indicador de los sensores de presión se ha añadido una opción especial ya que el número de trazas que se pueden llegar a representar en un solo gráfico es alto, cuarenta y cuatro, y ver un número tan alto de trazas en un mismo gráfico puede ser muy molesto e incluso dar lugar a error al no distinguir una traza con otra.

Para evitar esto se ha dispuesto de una serie de opciones que permitan seleccionar que traza ver o no ver en cada momento. Estas opciones se han elegido haciendo *click* con el botón derecho sobre el indicador de presión y a continuación seleccionar *create property node*. Una vez dentro del menú *property node* saldrán un montón de opciones, las que elegidas han sido: *Active plot* (figura 49) que junto con un control a su entrada permite seleccionar las trazas, también se ha dispuesto de una serie de condiciones para que no se puedan seleccionar más de cuarenta y cuatro trazas, nuestro límite para el número de sensores de presión, de tal forma que si se selecciona alguna traza por encima de la cuarenta y tres (cuarenta y cuatro trazas numeradas de la cero a la cuarenta y tres) se estaría haciendo mención a la cuarenta y tres.

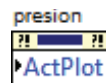


Figura 49: Active Plot

La otra opción o propiedad utilizada se llama: *visible* (figura 50). Esta opción permitirá ocultar o visualizar la traza seleccionada con la opción explicada anteriormente. A esta opción se le ha conectado un selector *true/false* de tal manera que cuando esté en *true* significará visualizar y *false* ocultar.

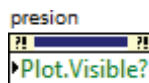


Figura 50: Visible

Lo que falta por explicar dentro del bucle *while* es la opción de registrar en un documento de texto las medidas provenientes de los sensores. Esta opción se encuentra dentro de una estructura *case* a cuyo selector se ha conectado un control para indicar que se quieren capturar los datos cuando este control valga *true* y que deje de escribir los datos o no escriba mientras valga *false*.

Para guardar las medidas de los sensores en un documento de texto se ha utilizado el VI de la figura 51:

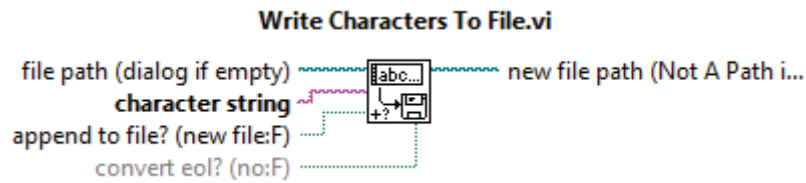


Figura 51: *Write Characters To File.vi*

Este VI escribe la cadena de caracteres introducida por la entrada *character string* en el documento indicado por la entrada *file path*.

Para indicarle la ruta del archivo donde se quieren escribir las medidas se ha utilizado la función de la figura 52:

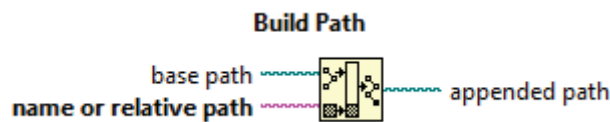


Figura 52: Función *Build Path*

Esta función permite crear una nueva ruta o path añadiendo el nombre del nuevo archivo. La ruta se la indicaremos por la entrada *base path* y el nombre por la otra entrada, *name or relative path*.

Se ha utilizado otra función con el fin de que el usuario solo tenga que indicar el nombre del archivo sin la necesidad de que incluya la extensión “.txt”. La función utilizada permite concatenar cadenas de caracteres por lo que en primer lugar se pondrá el nombre del archivo y a continuación se añadirá la extensión. La función está representada en la figura 53.

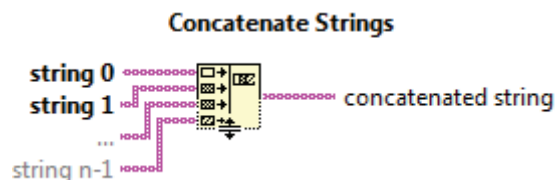


Figura 53: Función *Concatenate Strings*

Para proporcionarle al VI *Write characters to file* los datos que se quieren escribir se ha tenido que convertir el *array* donde están almacenadas las medidas ya que se trata de un

array de números y el VI *Write characters to file* necesita cadenas de caracteres. Para realizar esta conversión se ha utilizado la función de la figura 54.

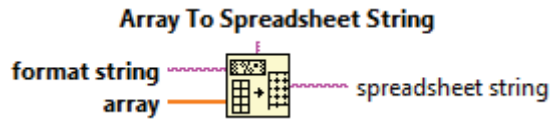


Figura 54: Función *Array To Spreadsheet String*

Se pondrá el *array* de medidas en la entrada *array* y por la entrada *format string* se indicará que se quieren datos reales (*float*) con tres decimales. A continuación se vuelve a usar la función *Concatenate Strings* para establecer la separación entre los caracteres de la cadena, en este caso se ha usado el tabulador.

La última parte del programa tiene lugar cuando el usuario pulsa el control llamado *terminar* que hace que se salga de este bucle *while* que se viene describiendo hasta el momento. Una vez que el usuario pulse ese pulsador se eliminan las tareas o enlaces que se han creado con cada una de las tarjetas que se han usado. El VI utilizado está representado en la figura 55.

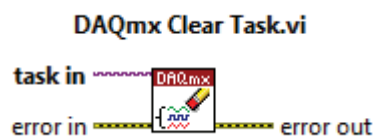


Figura 55: *DAQmx Clear Task.vi*

Se ha un VI como el de la figura 55 para cada tarjeta. Se han usado estructuras *case*, en las que se ha utilizado este VI, para no limpiar tareas o enlaces con tarjetas que no usadas. A la entrada *task in* se han conectado las salidas *task out* de los VI *DAQmx Read* de cada tarjeta y se ha hecho lo mismo con los respectivos *error out* que han sido conectados a las entradas *error in*.

Una vez realizado esto finaliza el contenido del segundo subdiagrama de la estructura *Flat Sequence* y con ello la explicación de toda la aplicación.

#### 3.1.4.2 Aplicación para adquirir datos de sensores y controlar una válvula

El diagrama de bloques de la aplicación que se va a explicar a continuación, que incluye el control de una válvula, es el mismo que se ha explicado anteriormente pero incluye más cosas para poder realizar ese control y algunas modificaciones.

La parte que hemos añadido para el control de la válvula se corresponde con parte de una aplicación ya realizada que se está usando en el laboratorio de Hidráulica de la Escuela Universitaria de Ingeniería de Caminos, Canales y Puertos.

La mayor parte del código que se ha incluido se encuentra dentro de una estructura *case* cuyo selector se ha conectado a un control para habilitar (*true*) o no la ejecución del control de la válvula, este control se llama *válvula*.

La parte que queda fuera de esa estructura *case* es la correspondiente con la creación de la tarea que enlaza con la tarjeta a la que está conectada la válvula. Para conectar la válvula se va a utilizar otro tipo de tarjeta, una tarjeta que permita enviar señales analógicas para así poder mandar órdenes a la válvula. Para crear la tarea de enlace con la tarjeta a la que está conectada la válvula se ha vuelto a usar el VI *DAQmx Timing* como se usó con las tarjetas a las que estaban conectados los sensores. Este VI esta dentro de una estructura *case* cuyo selector está controlado por el control que habilita o no el control de la válvula.

La modificación se ha realizado sobre el VI *calculos*. Como se comentó en la explicación del documento de Excel y que posteriormente se desarrollará, el control de la válvula requiere de dos parámetros importantes para funcionar, el correspondiente a la medición obtenida por un sensor de caudal conectado al puerto AI14 de la tarjeta número uno y una la salida de la válvula conectada al puerto AI13 de la tarjeta 1 mediante la cual informa del grado de abertura que tiene (su posición). Las unidades de estos dos parámetros han de estar en corriente por lo que se tendrán que coger antes de que pasen por el *Formula Node*.

Dentro del VI *calculos* lo primero que se ha usado ha sido una estructura *case* para saber si la medida que está llegando proviene de algún sensor de la tarjeta número uno, si es así, se utilizarán otras dos estructuras *case* cuyo selector provendrá de la comparación con el puerto para saber si el puerto es el número trece, sensor de caudal, o catorce, posición de la válvula. Si es alguno de los dos puertos nombrados se obtendrán sus medidas que provienen del *array* de medidas de la tarjeta número uno que ha sido multiplicado por mil. Estos dos valores saldrán por dos salidas que se han añadido al VI para poder usarlos en la estructura *case* donde se encuentra el control de la válvula.

Para resumir la parte del control de la válvula se empezará indicando que la válvula tiene dos modos de funcionamiento: control remoto y control manual, la válvula físicamente tiene un selector para ello. Si se pone la válvula en control manual seremos nosotros los que físicamente girando la llave que está junto a la válvula controlemos la apertura de la misma. Si se pone la válvula en control remoto seremos nosotros a través de LabVIEW los que controlemos la válvula.

Partiendo de que la válvula está en modo control remoto. Se tienen dos formas de controlar la válvula, se puede controlar la válvula indicando el caudal que queremos en litros por segundo o el grado de apertura que se desea para la válvula. Para seleccionar estas dos formas de funcionamiento se ha utilizado una estructura *case* cuyo selector está conectado a un control *true/false* llamado *Control automático del caudal*. Cuando el control entregue el valor *true* se controlará la válvula eligiendo el caudal que se desea y si entrega el valor *false* la apertura en grados.

Si se elige controlar la válvula a través de un caudal determinado se tienen dos opciones: elegir el caudal deseado a través de un control llamado *Q referencia* o utilizar un documento de texto que previamente se haya creado y rellenado donde estén especificados los caudales deseados para un tiempo determinado.

Estas dos opciones están gestionadas dentro de una estructura *case* con dos posibles opciones: cero o uno. Para elegir la opción cero o la uno se ha utilizado una *list box* donde se puede elegir *Caudal constante* (0) o *Hidrograma* (1) y que se ha conectado al selector de la estructura *case*. Si se elige la opción *Caudal constante* se tendrá que indicar a través del control *Q referencia* el caudal que deseado. Si se elige la opción *Hidrograma* se utilizará un documento de texto como se ha comentado anteriormente.

La estructura que debe seguir el documento de texto es la siguiente: en la primera columna se pondrá el tiempo en segundos para el cual se desea un caudal determinado y en la segunda columna, separada de la primera por un tabulador, el caudal en litros por segundos que se desea para el tiempo que hemos puesto en la primera columna. Para leer los datos de este documento se ha usado el VI representado en la figura 56.

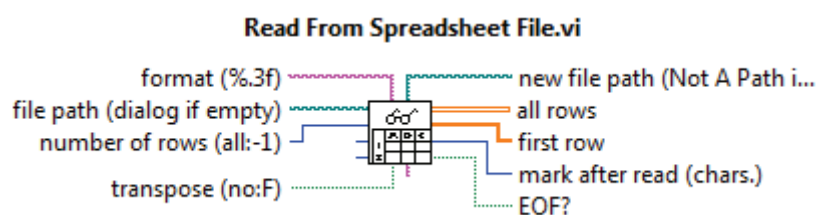
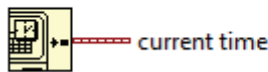


Figura 56: Read From Spreadsheet File.vi

Como parámetro de entradas se introducirá el *path* donde se encuentra el documento de texto por la entrada *file path* y se obtendrán, por la salida *all rows*, todas las filas del documento.

Para obtener la fila adecuada de las dos columnas usamos la función *Index Array*. Se utilizarán dos funciones *Index Array*, una de ellas para obtener filas de la primera columna y otra para las de la segunda columna. La elección de una u otra fila viene determinada por el tiempo que haya transcurrido, es decir, la primera vez se cogerá los valores de la primera fila y cuando haya transcurrido un tiempo igual al que se ha escrito en la segunda fila de la primera columna se saltará a la segunda fila y así sucesivamente. Para saber el tiempo que ha pasado usamos la función representada en la figura 57.

**Get Date/Time In Seconds**



**Figura 57: Función Get Date/Time In Seconds**

Esta función proporciona la hora actual. Se ha situado uno fuera del bucle *while* y otro dentro de la estructura *case* que controla la válvula. La operación realizada para saber el tiempo que ha transcurrido es restar la hora de la función que está fuera del bucle *while* a la hora del que está dentro de la estructura *case*. La primera vez que se ejecute el bucle *while* la función que está fuera del mismo dará un valor y este valor no se verá actualizado, visto desde dentro del bucle *while*, hasta que no se salga del mismo y se vuelva a ejecutar, cosa que solo sucedería si se detiene la aplicación y se vuelve a iniciar. La función que se encuentra dentro de la estructura *case* se ejecuta continuamente por lo que la hora estará actualizada, de esta manera se tendrá una referencia del tiempo que ha pasado para saber así si saltar o no a la siguiente fila del documento de texto.

La aplicación dispone de un control *true/false*, *Poner tiempo a cero*, que si se pulsa, *true*, permite volver a la primera fila del documento, esto es debido a que se toma como nueva referencia el tiempo en el que se ha pulsado este control.

La otra opción que falta por explicar es qué sucede cuando no se activa el *control automático del caudal (false)*. Si no se activa se podrá controlar la válvula según su posición, es decir, grados que la válvula debe abrirse. Dentro de controlar la válvula en grados se tienen dos opciones como sucedía con el caudal, o se seleccionan los grados de apertura o se utiliza un documento de texto en el cual se indique tiempo y grados como la opción *Hidrograma* para el caudal.

Al igual que sucedía antes con el caudal ahora se tiene una estructura case pero esta vez su selector va conectado a otro control *true/false* llamado *Control posición llave* que si está activado (*true*) permitirá elegir la abertura a través de un documento de texto que previamente se habrá rellenado como sucedía con la opción del *Hidrograma* para el caudal. La única diferencia que existe en el contenido de los dos documentos es que para esta opción los grados que la válvula debe abrirse estarán puestos en la tercera fila, es decir, poner el tiempo, pulsar dos veces el tabulador y a continuación poner los grados deseados. Si el control, *control posición llave*, está desactivado (*false*) se podrá controlar los grados que la válvula debe abrirse con otro control llamado *control abertura llave (°)* que incorpora una barra deslizante para seleccionar los grados.

Antes de enviar la correspondiente orden a la válvula hay que realizar una serie de conversiones ya que lo que se tiene hasta ahora es o un caudal en litros por segundo o una abertura de la válvula en grados.

Si se decide controlar la válvula a través de los grados que la válvula debe abrirse primero se realiza la división de los grados deseados entre nueve y a continuación enviar el resultado a la válvula.

Si se controla la válvula eligiendo un caudal determinado, primero se pasará este valor a través de un VI llamado *PIDHider2* (figura 58) que se trata de un sistema de control PID (Proporcional Integral y Derivativo) que además tiene en cuenta el valor proveniente de sensor de caudal y la salida que proporciona la válvula para elegir el valor adecuado a enviar a la válvula, este valor le llegará al VI a través de la entrada *Q referencia*.

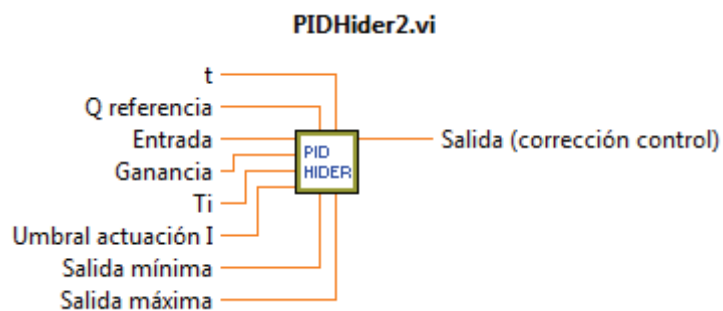


Figura 58: PIDHIDER2.vi

Una vez se tenga el valor que se va a enviar a la válvula se usará el VI de la figura 59 que permite enviar valores a la válvula.

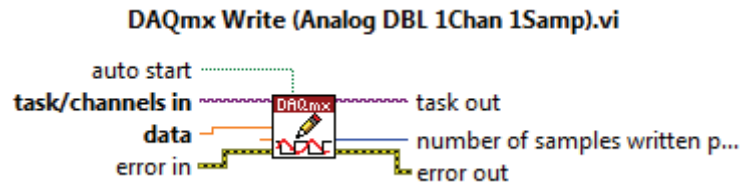


Figura 59: DAQmx Write (Analog DBL 1Chan 1Samp).vi

Este VI recibe por la entrada *task in* la salida *task out* del VI *DAQmx Timing* asociado al puerto donde está conectada la válvula y por la entrada *data* recibe el valor que se quiere enviar a la válvula.

De la misma forma que sucedía con las tareas asociadas a las tarjetas donde están conectados los sensores, una vez que se pulse el control *terminar* hay que eliminar la tarea que se ha creado con el VI *DAQmx Clear Task*. Este VI se encuentra dentro de una estructura *case* cuyo selector está controlado por el control *valvula* para que solo se ejecute si se ha decidido usar la válvula.



### 3.1.5. Manuales de usuario

A continuación se va a describir la parte con la que tendrá que interactuar el usuario para poder controlar las dos aplicaciones realizadas, esta parte se corresponde con los paneles de control de dichas aplicaciones.

#### 3.1.5.1 Panel de control de la aplicación sin válvula

El archivo que tenemos que abrir para ejecutar el programa es el siguiente: *adquisicion\_sin\_valvula.vi*.

En la figura 60 se puede ver el panel frontal de esta aplicación.

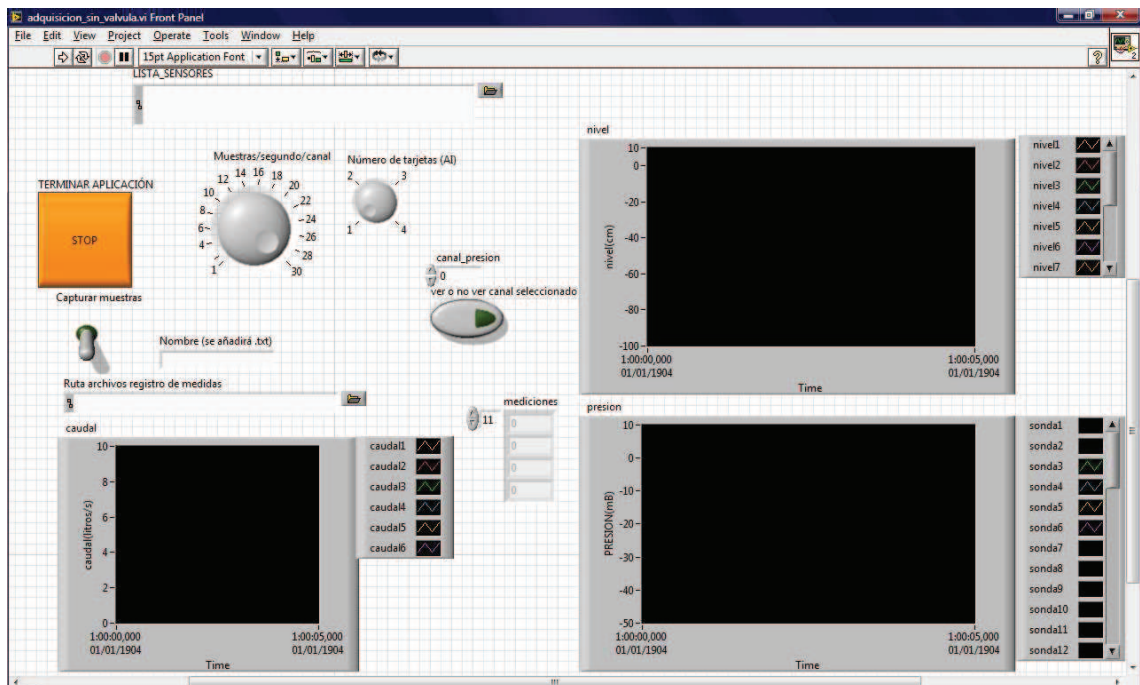


Figura 60: Front Panel aplicación sin control de válvula

Para empezar a usar la aplicación hay que elegir tres parámetros sin los cuales la aplicación no funcionaría:

1º-El documento Excel donde se encuentra la lista de sensores con sus respectivos parámetros.



Figura 61: Selector de la ruta de Documento de Excel

La figura 61 representa donde se tiene que seleccionar el path donde se encuentra el documento de Excel, pulsando en la carpeta.

2º-La muestras por segundo por canal (figura 62), cuyo valor máximo será treinta.



Figura 62: Control Muestras/Segundo/canal

3º-El número de tarjetas, *Analog Input* (AI), que se tengan conectadas para adquirir datos. La figura 63 muestra este control.

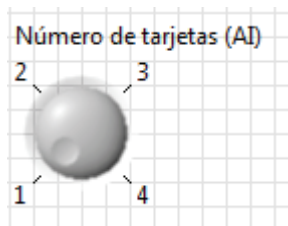


Figura 63: Control Número de tarjetas (AI)

Si se olvida seleccionar el número de tarjetas que se van a utilizar por defecto se estará seleccionando una tarjeta

Una vez elegidos estos parámetros se lanza la aplicación pulsando el botón *run* (figura 64).

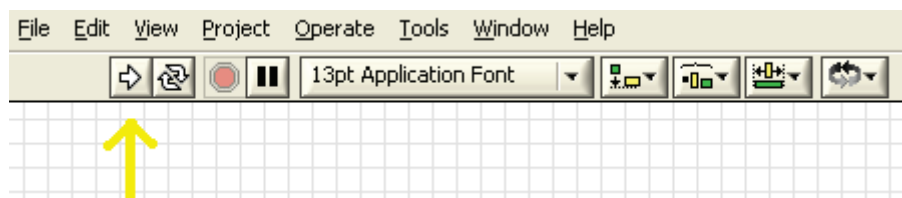


Figura 64: Botón Run

En los tres gráficos (figura 65) que se han utilizado aparecerán los valores que llegan de los sensores. Se dispone de un gráfico para cada uno de los tipos de sensores que se pueden utilizar, uno para sensores de presión en el cual se representarán presiones en milibares, otro para sensores de nivel representando alturas en centímetros y otro para sensores de caudal utilizando unidades de litros por segundo para representar caudales. En todos los gráficos el eje Y representa la hora actual.

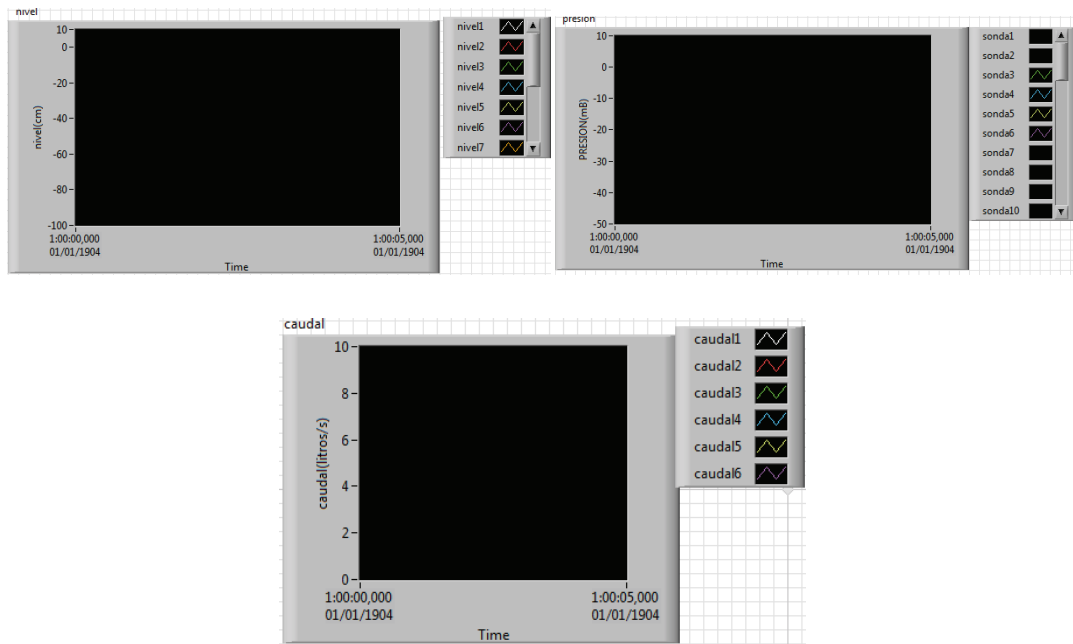


Figura 65: Indicadores gráficos de nivel, presión, y caudal

La escala en el eje vertical puede ser modificada en cualquier momento sin la necesidad de detener la aplicación, solo será necesario hacer *click* con el botón izquierdo sobre uno de los valores extremos y modificarlo.

Igual sucede con las características de las trazas, solo será necesario hacer *click* con el botón derecho sobre la traza en la leyenda para modificar el color de la traza, su estilo y grosor. En la figura 66 se puede ver el símbolo de una traza y el menú de opciones que aparece al hacer *click* con el botón derecho sobre el símbolo de una traza.



Figura 66: Trazas y opciones de la misma

Respecto al gráfico de presiones, como se explicó anteriormente, se ha incorporado una opción para poder visualizar solo las trazas deseadas ya que representar todas las trazas de todos los sensores de presión que se pueden conectar a la vez sería bastante lioso. Para ello usamos los controles representados en la figura 67.

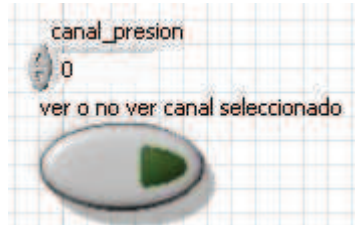


Figura 67: Opciones para ver o no ver trazas

Con el selector *canal\_presion* se selecciona un canal asociado a una traza y con *ver o no ver canal seleccionado* se elige si se quiere ver (figura 68) o no ver (figura 69) la traza del canal seleccionado.



Figura 68: No ver



Figura 69: Ver

Cuando se elija ver (figura 71) o no ver (figura 70) una traza la leyenda cambiará de aspecto.

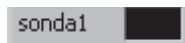


Figura 70: Trazas ocultas

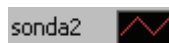


Figura 71: Trazas visibles

Ejemplo:

Como podemos ver en la figura 72 tenemos todas las trazas invisibles menos las de las sondas dos, tres y cuatro.

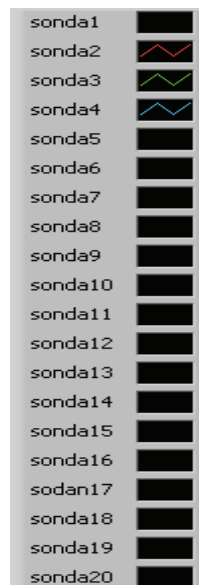


Figura 72: Trazas

En la tabla 6 se resume cómo se ha llegado a esa situación.

sonda	canal_presión	Ver(encendido) o no ver(apagado)
1	0	Apagado
2	1	Encendido
3	2	Encendido
4	3	Encendido
5	4	Apagado
6	5	Apagado
7	6	Apagado
.	.	.
.	.	.
20	19	Apagado

Tabla 6: Ejemplo situación de trazas

También se dispone de un *SCROLL* (figura 73) para las gráficas de nivel y presión para deslizarlos sobre la leyenda de los sensores que se están graficando.



Figura 73: *SCROLL*

Si se quieren registrar los datos adquiridos en un documento de texto solo será necesario indicar el nombre del archivo (figura 74) que se va a generar, la ruta donde se generará el archivo (figura 74) y finalmente pulsar sobre el selector *capturar muestras*. Este selector cambiará de apariencia cuando se pulse (figura 75 y 76). Cuando queramos detener la captura lo volveremos a pulsar.

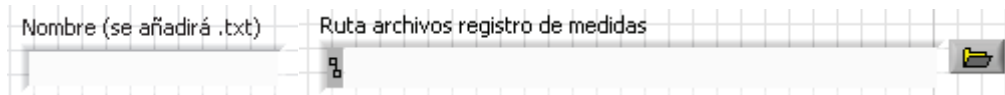


Figura 74: Nombre y ruta del archivo de registro de medidas



Figura 75: No capturar



Figura 76: Capturar

Desde el panel frontal también se puede visualizar el valor proveniente de los sensores de otra forma. Para ello se dispone de un indicador con el que se puede ver de forma numérica el valor entregado por los sensores. Este indicador (figura 77) se corresponde con el *array mediciones*.

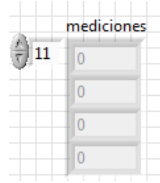


Figura 77: Indicador *array mediciones*

En la figura 77 se tiene seleccionada la posición número once del *array* que se corresponderá con el canal AI11 de la tarjeta número uno, el valor de la posición once del *array* aparecerá justo a la derecha, es decir, el primer valor que aparece en lista mediciones, el siguiente sería la posición doce y así sucesivamente (figura 78).

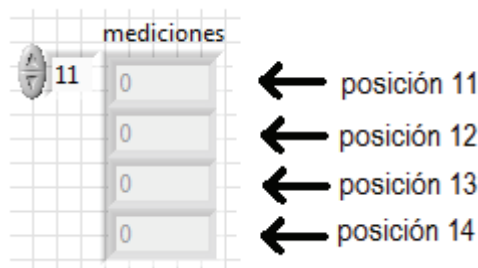


Figura 78: Indicador *array mediciones* explicado

Si se quiere modificar el número de muestras por segundo por canal se tiene que parar la aplicación pulsando el pulsador *TERMINAR APLICACIÓN* (figura 79), a continuación se modificará a nuestro gusto el número de muestras por segundo por canal y se volverá a lanzar la aplicación pulsado el botón *run* (figura 64).



Figura 79: Control *TERMINAR APLICACIÓN*

### 3.1.5.2 Panel de control de la aplicación con válvula

Todo lo descrito anteriormente sirve para esta aplicación salvo algunas modificaciones que se han realizado para poder usar la válvula.

El archivo que tenemos que abrir para ejecutar esta aplicación es el siguiente: *adquisición\_con\_valvula.vi*.

En la figura 80 se puede ver el panel frontal de esta aplicación.

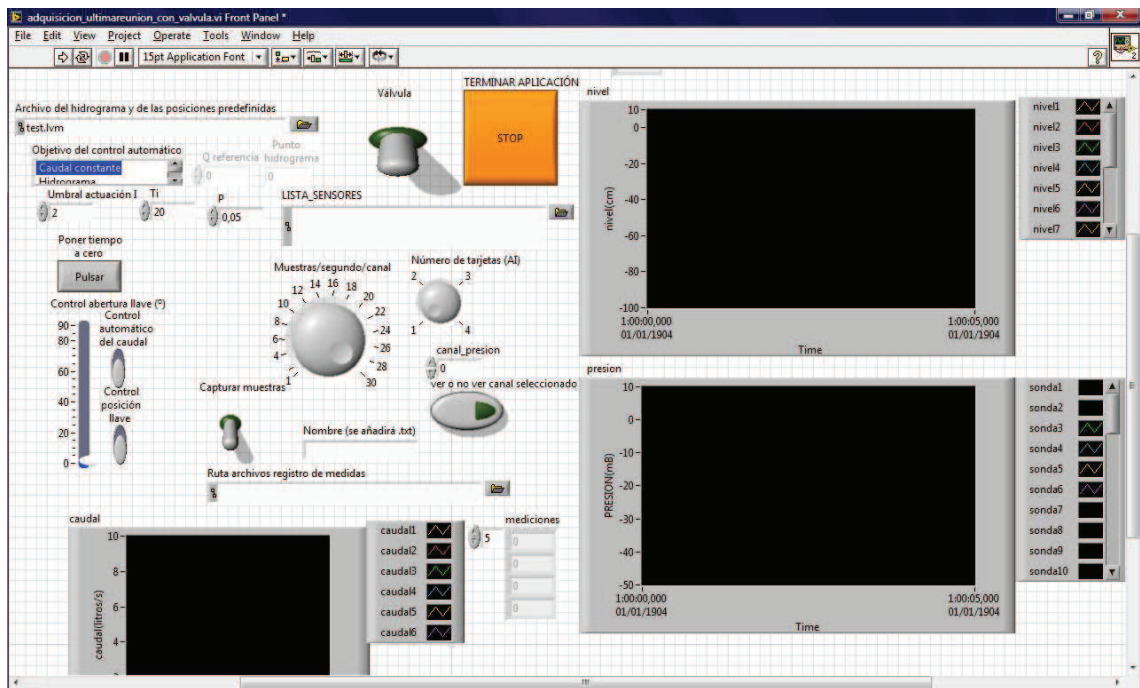


Figura 80: *Front Panel* aplicación con control de válvula

Para empezar a usar la aplicación hay que elegir los tres parámetros explicados anteriormente sin los cuales la aplicación no funcionaría.

A continuación se listan una serie de detalles a recordar y las modificaciones introducidas en la aplicación con respecto a la anterior en lo que al panel de control se refiere.

Recordar que:

- En este programa se cuenta con la presencia de una válvula cuyo terminal de salida ha de ser conectado al puerto número 13 (AI13) de la tarjeta número uno. Para el resto de tarjetas de adquisición ese puerto queda libre para sensores de presión.



-La válvula va de la mano de un sensor de caudal conectado al puerto 14 (AI14) de la tarjeta número uno.

-El control de la válvula va en una tarjeta aparte, *Analog Output (AO)*, por lo que el número máximo de tarjetas conectadas para adquirir datos, *Analog Input (AI)* será de tres (figura 63).

En función de lo seleccionado se remite al usuario al apartado de conexionado relacionado con el número de tarjetas conectadas para adquirir datos de sensores. Si no se quiere controlar válvula el funcionamiento del programa será el mismo que el software que no lleva válvula y si se quiere controlar la válvula se tendrá que tener en cuenta más aspectos que se describen a continuación.

Cambios en el panel de control:

-Selector de válvula: servirá para seleccionar si se quiere controlar la válvula. Este control cambiará de apariencia cuando sea pulsado (figuras 81 y 82).



Figura 81: Control válvula activado



Figura 82: Control válvula desactivado

-El resto de controles añadidos en el panel de control para el control de la válvula se encuentran en la parte izquierda del mismo. En la figura 83 se pueden ver estos controles.

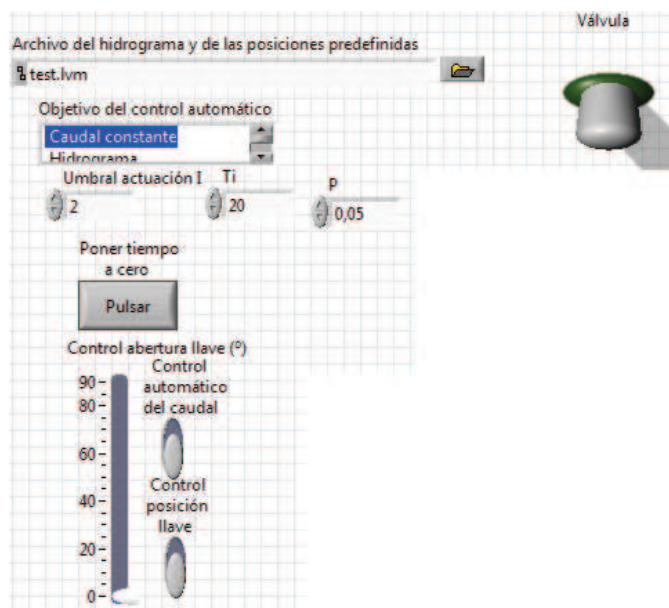


Figura 83: Controles para controlar la válvula

Si se activa el control llamado *Control automático del caudal* (figura 84), cambiando éste de apariencia, se podrá controlar la válvula de dos formas distintas que se seleccionan con el control llamado *Objetivo del control automático* (figura 85).

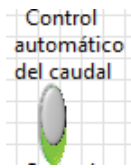


Figura 84: Control automático del caudal activado

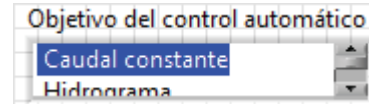


Figura 85: Objetivo del control automático

Si se selecciona *Caudal constante* aparecerá a la derecha un control llamado *Q referencia* (figura 86) en el cual se podrá elegir el caudal deseado en litros/segundo.

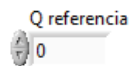


Figura 86: Control *Q referencia*

Si se elige la otra opción, *Hidrograma*, la válvula se controlará en función del documento de texto en el cual se hayan introducido los parámetros de tiempo y caudal deseados. Para elegir la ruta donde se encuentra ese documento se tiene que seleccionar pulsando en la carpeta de la figura 87:

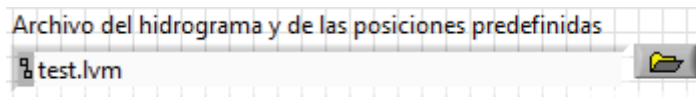


Figura 87: Selección del documento para el control de la válvula

Si no se activa el control *Control automático del caudal* se podrá controlar la válvula según su posición, es decir, grados que se quiere que se abra. Dentro de controlar la válvula en grados se tienen dos opciones como sucedía con el caudal, o se seleccionan los grados que la válvula ha de abrirse o se utiliza un documento de texto en el cual se indique el tiempo y los grados como la opción *Hidrograma* para el caudal.

Para seleccionar control manual o usar el documento de texto se tiene que utilizar el control de la figura 88:

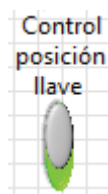


Figura 88: Control posición llave activado

Como se presenta en la figura 88 el control está activado, lo que indicaría que se está seleccionando usar el documento de texto. Si se usa esta opción se tiene que indicar la ruta donde se encuentra el documento de texto (figura 87).

Si no se activa el control: *Control posición llave* se elegirá a través del control de la figura 89 la abertura en grados deseada

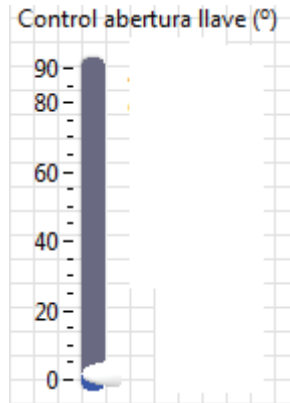


Figura 89: Control abertura llave (°)

Una vez elegidos estos parámetros lanzamos la aplicación pulsando el botón *run* (figura 64).

Tanto para la opción del *Hidrograma* como para la del control por documento de texto de la posición se tiene el pulsador de la figura 90 para que, una vez se haya pulsado, se vuelva a empezar por la primera fila del documento de texto.

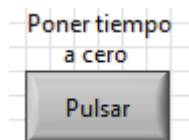


Figura 90: Poner tiempo a cero

### 3.1.6 Pruebas realizadas a sensores que se van a utilizar en experimentos

Hasta el momento se han conectado veinte sensores de presión modelo XA-300 para realizar un experimento que no requiere del uso de una válvula por lo que se ha utilizado la distribución de la tabla 1. Se han utilizado dos tarjetas NI-9208 ya que en una tarjeta, según la distribución utilizada, solo se pueden conectar hasta once sensores de presión. Estos sensores tienen una rosca en su parte superior la cual ha sido roscada a una pieza especial (figura 91) que se utilizará para el experimento.



Figura 91: Pieza y lugar del experimento

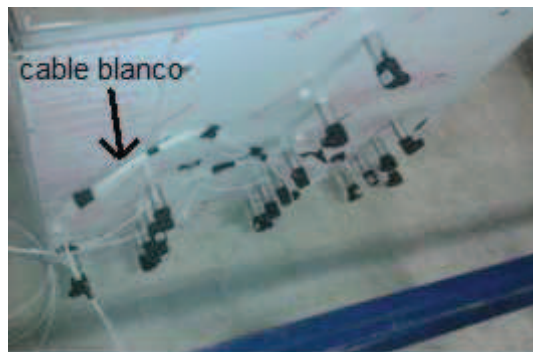


Figura 92: Pieza con los sensores roscados

Para realizar el cableado se ha utilizado un cable cuya parte exterior es de color blanco como se puede ver en la figura 92. Este cable contiene en su interior cuatro cables de colores verde, rojo, amarillo y blanco. Con estos cuatro cables se han conectado dos sensores ya que por sensor se necesitan dos cables.

Para otro experimento también se han conectado cuatro sensores de presión modelo XA-700 (figura 93).



**Figura 93: Sensores XA-700**

Todos los sensores de presión llevan una etiqueta en la cual hay escrito un número que los identifica.

Todos estos sensores han sido sometidos a una serie de pruebas para comprobar su correcto funcionamiento. Estas pruebas han consistido en someter a estos sensores a una serie de presiones que se encuentran dentro de sus márgenes de calibración. Para realizar estas pruebas se ha utilizado el tubo de la figura 94 que ha sido fabricado por los técnicos del laboratorio de hidráulica. Las pruebas consisten en que los sensores midan la presión que ejerce una columna de agua sobre ellos. La relación existente entre presión y columna de agua es la siguiente:

1 milibar de presión = 1 cm de columna de agua.

Para saber la cantidad de agua que echamos en el tubo se ha pegado una cinta métrica y para colocar los sensores se han colocado unas roscas en el fondo del tubo.



**Figura 94: Tubo para realizar pruebas**

Los niveles de agua a los que han sido sometidos los sensores van desde 0cm hasta 1m de agua con incrementos de 10cm. Una vez alcanzado el nivel de agua deseado se ha utilizado la aplicación de LabVIEW para capturar datos durante un periodo de tiempo. Una vez que se tengan esos datos en un documento de texto se llevarán a un documento de Excel y se calculará la media de todos ellos. A continuación sabiendo el nivel de agua aplicado y el valor entregado por los sensores se calcula el error restando al nivel de agua el valor obtenido del sensor.

Hay que tener en cuenta que el metro tiene su nivel cero de agua a la altura del fondo del tubo, a esta altura quedará la punta de la rosca del sensor quedando un centímetro por debajo la membrana a partir de la cual se encuentra el sistema de medición del sensor por lo que realmente siempre se estará sometiendo al sensor a aproximadamente un centímetro más de columna de agua de la lectura que se puede realizar con el metro fijado al tubo. Una vez se hayan calculado los errores para cada uno de los niveles aplicados se hará la media de estos. Las tablas 7 y 8 muestran la media de estos errores para cada sensor probado:

Sensor número	Media de error (mB o cm)	Sensor número	Media de error (mB o cm)
1	0,558985	11	-2,29366
2	-0,955	12	1,181749
3	1,277722	13	-0,1755
4	-2,3283	14	-2,09122
5	-2,00058	15	1,64616
6	0,996844	16	-1,55372
7	-1,74057	17	-1,8175
8	1,283036	18	-1,34112
9	1,525344	19	-0,36392
10	0,938025	20	-0,58563

Tabla 7: Media de errores sensores XA-300

Sensor	Media de error (mB o cm)
1	0,104447
2	0,60886
3	-2,51837
4	-0,43428

**Tabla 8: Media de errores sensores XA-700**

Según la hoja de características de los sensores estos tienen un error menor o igual al 0,3 % de su fondo de escala. Los sensores probados tienen un rango de medición de -300 a 300mB por lo que su fondo de escala es de 600mB. El 0,3 % de 600mB es 1,8mB que equivale a 1,8 cm de columna de agua sobre el sensor. Viendo el error permitido se ve que prácticamente todos lo cumplen dándose por valido el buen funcionamiento de éstos.

Este estudio puede ser meramente informativo y de comprobación de que los sensores funcionan correctamente o utilizarse como valor a introducir en el documento de Excel en la columna de referencias.

Respecto a los sensores de nivel solo se ha podido probar uno de ellos ya que no había más unidades. El modelo probado fue el sensor LU05 que tiene un rango de medición de 5 a 120cm. Primero se hicieron pruebas para comprobar que detectaba correctamente la altura a la que se encontraba y a continuación se utilizó la referencia introducida a través del documento de Excel, indicando la altura a la se situaba, para detectar niveles de agua.

Los sensores de caudal no se probaron ya que aún no se sabía que modelos se iban a adquirir.

Respecto a la válvula decir que por falta de tiempo no llegó a probarse con la aplicación realizada pero con la aplicación que estaba realizada con anterioridad funcionaba correctamente.



## 3.2 Láser y motor

### 3.2.1 Introducción

Se parte de una aplicación realizada con LabVIEW que permite controlar el funcionamiento de un láser y un motor (figura 95). La instalación de los dos dispositivos ya estaba realizada. El objetivo de controlar estos dos dispositivos con la aplicación es la de caracterizar presas realizadas a escala y detectar posibles roturas a partir de la realización de imágenes en tres dimensiones creadas con los datos obtenidos con el láser. El motor permitirá al láser, al que está unido, moverse a lo largo de una pasarela (figura 96) para así examinar la superficie de la presa que se vaya a estudiar.

El objetivo de esta parte del proyecto será detectar y corregir una serie de errores que se dan en la aplicación.

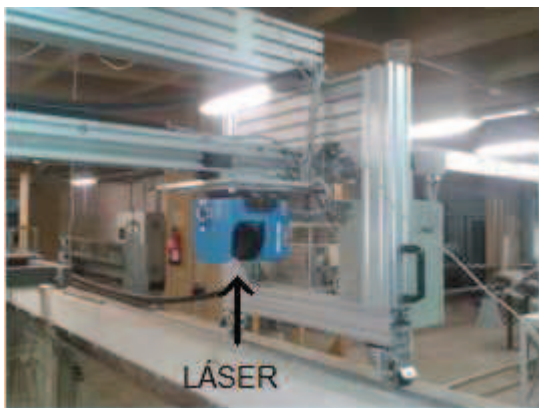


Figura 95: Láser y motor



Figura 96: Pasarela

### 3.2.2 Aplicación

El VI principal se encuentra en la siguiente ruta: Laser\_V4/LaserCamino.vi

El panel de control de la aplicación nos permite hacer diferentes pruebas indicándole las características del mismo. Se pueden realizar acciones que solo afecten a uno de ellos, láser o motor, u ambos.

La aplicación está gestionada mediante la utilización de eventos asociados a una serie de controles del panel de control que permiten realizar diferentes tareas.

El diagrama de bloques tiene una estructura de eventos en la cual se gestionan los diferentes eventos que se pueden generar. Esta estructura espera a que se dé un evento



para gestionarlo, si en un tiempo determinado, indicado a través del reloj de arena que se ve en la figura 97, no se da ninguno de los eventos asociados a la estructura se ejecutará la gestión de un evento llamado *Timeout*.

Esta estructura de eventos se encuentra dentro de un bucle *while* por lo que si se cumple el tiempo que da lugar a la gestión del evento *Timeout* se volverá a esperar.

Los eventos que se gestionan en esta estructura vienen asociados al cambio de valor (*true*, *false*) de unos controles situados en el panel de control. Una vez que se pulse uno de esos controles que da lugar a un evento lo primero que se hace es deshabilitar el resto de controles para evitar así que se dé otro evento mientras se está gestionando otro. Una vez se hayan realizado todas las tareas relacionadas con el evento se volverán a habilitar esos controles.

Para realizar las opciones de habilitar o deshabilitar estos controles se utilizan propiedades como las explicadas para visualizar las trazas de un gráfico.

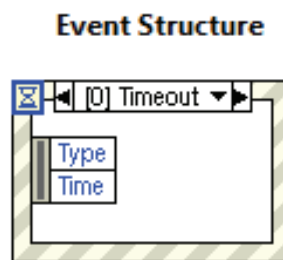


Figura 97: Event Structure

Los controles del panel de control (figura 98) que generan los eventos que son controlados por esta estructura son: *INICIALIZAR*, *HOMING*, *MOVER A*, *CONFIGURAR*, *SCAN PERFIL*, *SACAN 3D*.

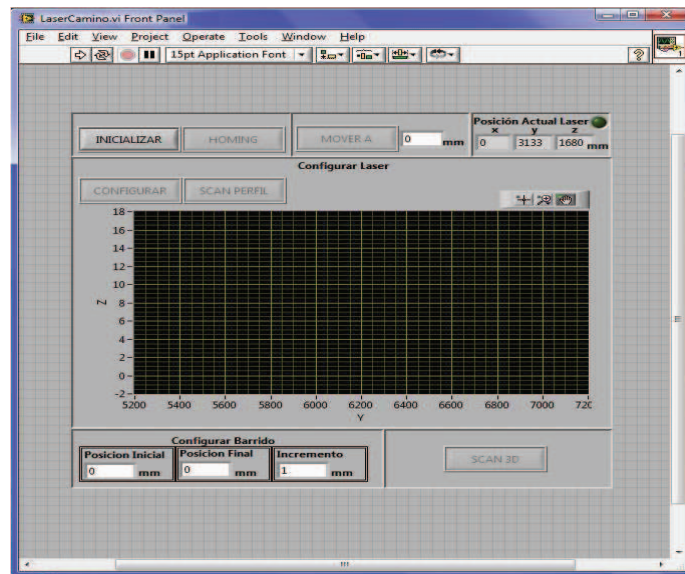


Figura 98: Front Panel LaserCamino.vi

Al arrancar la aplicación todos los controles se encuentran deshabilitados pudiéndose únicamente pulsar el control llamado *INICIALIZAR* para confirmar las comunicaciones entre el motor y el láser con el ordenador. La mayoría de los parámetros de la comunicación están ya predefinidos (*BaudRate*, nombre del dispositivo, interface...) y lo que habrá que comprobar será el puerto del ordenador al que están conectados cada uno.

Ambos dispositivos tiene un puerto serie con interfaz RS-232 para comunicarse. Para poder conectarlos a un ordenador moderno se ha utilizado un dispositivo que permita modificar el interfaz RS-232 a USB.

Respecto al motor se pueden realizar las siguientes operaciones:

- Llevar al motor a la posición inicial. Para ello el programa tiene un control llamado *HOMING*.
- Llevar al motor a una posición deseada. Para ello el programa tiene un control numérico en el cual se escribirá la posición deseada en milímetros y un control que se ha de pulsar para que el motor se mueva donde le hayamos indicado, *MOVER A*.

Respecto al láser:

-Se podrá obtener un perfil del máximo rango que puede ver el láser, 0-180°. Para ello solo habrá que pulsar un control que se llama *SCAN PERFIL*.

Conjuntamente se puede realizar la siguiente operación:

-Si se pulsa un control llamado *CONFIGURAR* aparecerá un menú en la pantalla en el que se introducirán las características del barrido deseado para el láser, estos son:

-Ángulo inicial.

-Ángulo final.

-Incremento de grados: 0,25°, 0,5° o 1°.

-Número de muestras. Este parámetro hace referencia al número de veces que queremos que realice un muestreo.

-Para el motor se tienen una serie de controles numéricos en los que se introducirán los siguientes parámetros:

-Posición Inicial

-Posición Final

-Incremento

Una vez introducidos estos parámetros se pulsará el control *SCAN 3D*.

A partir de que se pulse el control *SCAN 3D* se abrirá un VI que se utiliza dentro del programa principal y que tiene el mismo nombre del control. A continuación el motor se moverá desde la posición inicial introducida hasta la posición final con el incremento introducido. Antes de que el motor se mueva a la siguiente posición el láser entrará en juego realizando tantos barridos como se haya introducido a través del parámetro “número de muestras” entre los ángulos introducidos a través de “ángulo inicial” y “ángulo final” con el incremento que se haya indicado con “incremento de grados”. Cuando el láser realice un barrido los datos obtenidos se representarán en un gráfico en tres dimensiones que se encuentra en el VI *Scan 3D*.

Finalmente cuando el motor llega a su posición final se podrá seleccionar si queremos salir del VI *Scan 3D* y volver al VI principal para realizar otra prueba o guardar y salir, para estas dos opciones existen dos controles. Si se selecciona guardar y salir aparecerá en la

pantalla un menú en el cual se podrá elegir el nombre y destino de un archivo *winrar* donde se guardarán los datos obtenidos en todos los barridos expresados a través de las coordenadas "X", "Y" y "Z". Si se elige salir se retornará al VI principal sin que los datos sean guardados.

### 3.2.3 Detección y corrección de errores

#### 3.2.3.1 Detección y corrección de errores de la aplicación

Se realizaron numerosas pruebas para ver cuándo, cómo y dónde fallaba la aplicación. Las pruebas se diferenciaban en la distancia que debía de recorrer el motor y los grados que debía de barrer el láser.

Tras numerosas pruebas con resultados satisfactorios la aplicación dejó de funcionar, apareció un mensaje con letras blancas y fondo azul en la pantalla que enseguida desaparecía (figura 99), la pantalla se quedaba en negro y había que reiniciar el ordenador. Otras veces el ordenador se quedaba bloqueado donde estaba y no respondía a ningún tipo de señal mandada desde el ratón o el teclado o mostraba un mensaje en la pantalla como los de figura 100.

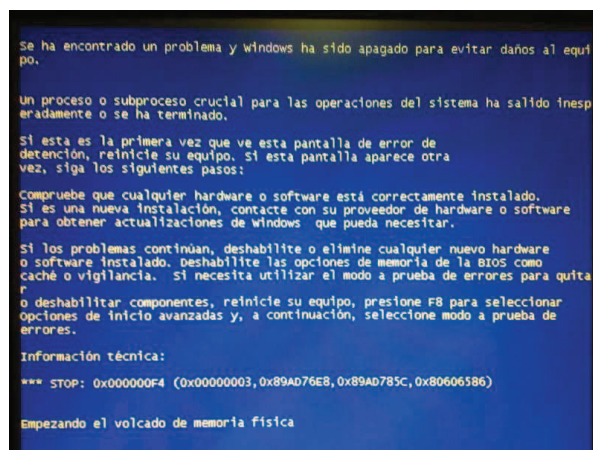


Figura 99: Pantalla azul con letras blancas

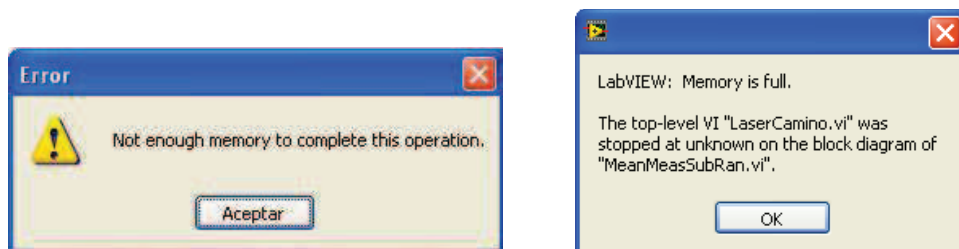


Figura 100: Mensajes de error

Vistos los problemas se decidió tomar dos vías:

- Ponerse en contacto con el servicio técnico de *National Instruments*.
- Buscar en la aplicación las partes que no hicieran un buen uso de la memoria.

#### 3.2.3.1.1 Memoria Virtual

El servicio técnico de *National Instruments* señaló que este comportamiento aparece típicamente cuando se está tratando con grupos de datos muy grandes en LabVIEW. La razón más común de este error no es que el ordenador se haya quedado completamente sin memoria. Más bien LabVIEW genera este error si no puede alojar la información en bloques de memoria contigua. Cuando se trabaja con datos muy grandes por ejemplo un *array* de 2000 x 2000 dobles, este ocuparía aproximadamente 32MB de información. LabVIEW puede necesitar en ocasiones alojar este espacio varias veces dependiendo de su programa puede tenerlo en más de una copia de estos datos.

La solución propuesta fue aumentar el tamaño de la memoria virtual siguiendo los siguientes pasos:

1. Desde el Panel de control hacer doble *click* en **Sistema**.
2. Hacer *click* en la pestaña de **Avanzado** y después en configuración en la sección de configuración.
3. A continuación hacer *click* en la pestaña de **Avanzado** y después *click* en el botón de cambiar en la sección de **Memoria Virtual**.
4. Puede fijar un valor inicial de memoria Virtual y un Valor máximo modificando los números y presionando en botón de fijar.
5. Reiniciar el ordenador para que los cambios tengan efecto.

Una vez realizados estos cambios, aumentando la memoria virtual hasta en 10Gb, el programa parecía que funcionaba correctamente pero de nuevo volvía a fallar de vez en cuando mostrando el mismo error.

### 3.2.3.1.2 Modificación de la aplicación

Por nuestra parte se detectó una parte de código en la cual no se realizaba una gestión adecuada de la memoria. Esta parte del código se encontraba en la parte donde se guardaban los datos provenientes del láser, coordenadas “X”, “Y” y “Z”. El problema estaba en que no se reservaba el espacio de memoria necesario sino que no reservaba espacio y según éste iba siendo necesario se producía el efecto comentado por el servicio técnico de *National Instruments*, al necesitarse más memoria se copiaban los datos existentes a otras partes de la memoria del ordenador y ésta cada vez se iba llenando más y más.

La solución adoptada para ello fue la creación de un VI que en función de la prueba que se fuera a realizar reservara un espacio de memoria a tal efecto.

El VI recibe como parámetros de entrada los datos introducidos por el usuario en relación al motor y al láser, estos son:

-Del láser:

-Ángulo inicial.

-Ángulo final.

-Incremento de grados.

-Del motor:

-Posición Inicial

-Posición Final

-Incremento

El VI se muestra en la figura 101 y se llama *matrices\_datos.vi*:

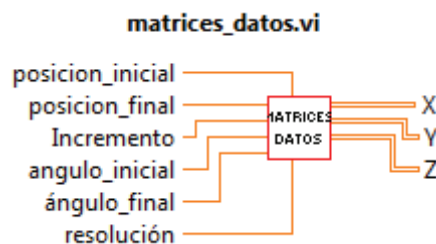


Figura 101: *matrices\_datos.vi*

Una vez se tengan todos los datos disponibles en la entrada del VI éste se ejecutará.

Las salidas que obtenidas de este VI son tres *arrays* de dos dimensiones cada uno en los cuales se irán guardando los datos provenientes del láser de las tres dimensiones espaciales “X”, “Y” y “Z”. La coordenada “X” se corresponde con la dirección en la que se mueve el motor. La “Z”, perpendicular a la “X”, y mirando hacia el suelo y al techo. La “Y” es la otra coordenada restante perpendicular a las coordenadas “X” y “Z”.

El número de filas que tendrán los *arrays* viene determinado por el número de pasos que da el motor. Este valor se obtiene realizando la siguiente operación:

$$filas = \frac{Posición\ final - Posición\ inicial}{Incremento} + 1$$

En este caso no es necesario chequear si la posición final introducida es menor que la posición inicial ya que todo el contenido del VI *Scan 3D*, donde se encuentra nuestro VI *matrices\_datos*, está dentro de una estructura case cuyo selector está conectado a una función *Greater Or Equal?* (figura 102) que ya chequea los parámetro introducidos. Si esta función entregara el valor *false* significaría que los parámetros están introducidos correctamente y se ejecutaría todo el contenido del VI *Scan 3D*. Si entregara el valor *true* significaría que la posición inicial es mayor que la final, se mostraría un mensaje en pantalla indicando este error y se tendría que volver a introducir todos los parámetros.

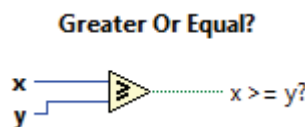


Figura 102: Función *Greater Or Equal?*

El número de columnas viene determinado por el barrido que realiza el láser en cada posición a la que se mueve el motor.

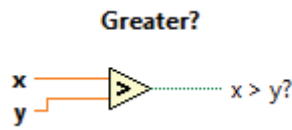
Para calcular el número de columnas se realiza la misma operación realizada para las filas pero con los parámetros del láser:

$$Columnas = \frac{Ángulo\ final - Ángulo\ inicial}{Resolución} + 1$$

En el caso de que, por error, el usuario haya introducido un valor para la posición inicial mayor que la para la posición final se invertirán estos valores, para ello se ha utilizado una estructura case cuyo selector está conectado a la función *Greater?* (figura 103) que valdrá *true* cuando las posiciones se hayan introducido correctamente y *false* cuando la



posición inicial sea mayor que la final. En este último caso, *false*, en vez de restar a la posición final la inicial se hará al revés.



**Figura 103: Función Greater?**

El valor que proporcionado a la entrada resolución no se corresponde con las resoluciones posibles, es decir, el usuario elige una resolución en función del rango de ángulos que quiera barrer. En función del rango elegido llegarán los valores representados en la tabla 9.

Rango de ángulos	Resolución	Valor que nos llega
40° - 140°	1°	0
40° - 140°	0,5°	1
40° - 140°	0,25°	2
0° - 180°	1°	3
0° - 180°	0,5°	4

**Tabla 9: Valores asociados a cada resolución**

Ejemplo: si el rango de valores que se quiere barrer está, por ejemplo, entre los ángulos 0 y 180, y se elige una resolución de 0,5° el valor recibido por la entrada resolución será el 4.

Sabiendo esto se han realizado una serie de comparaciones con la función *Equal?*, *Or* y *Select* para saber qué resolución es la que se ha seleccionado. Estas comparaciones se pueden ver en la figura 104.

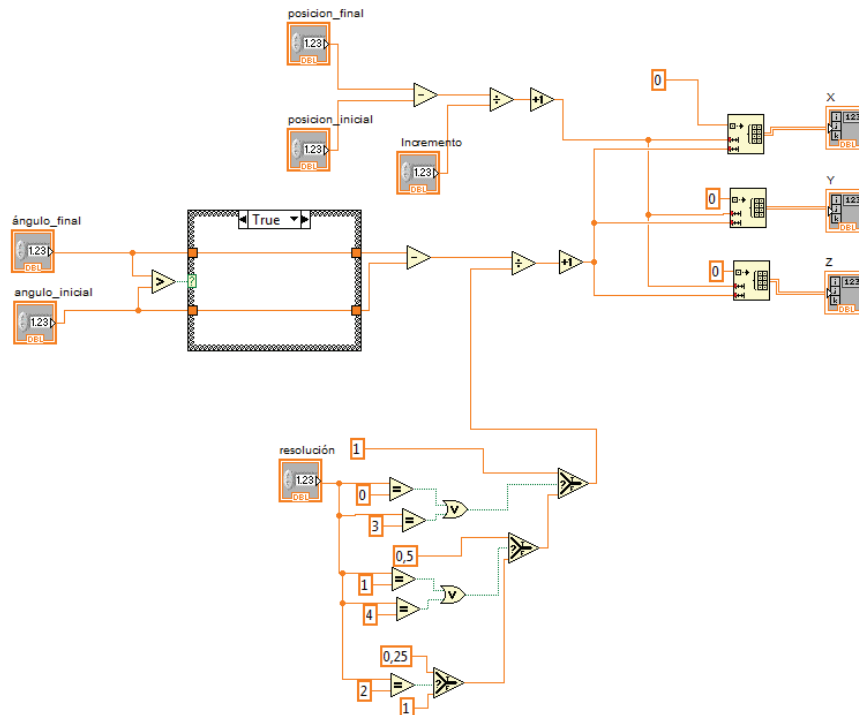


Figura 104: Diagrama de bloques de matrices\_datos.vi

Una vez se tenga el número de posiciones de filas y columnas necesarias se crearán los *arrays*. Para ello se utilizan tres funciones *Initialize Array*, una para cada coordenada, en las cuales se indicará el tamaño del *array* a través del número de filas y columnas calculadas anteriormente y el dato con el que se rellenara cada una de las posiciones que será 0. Estos *arrays* llegan a unos *shift register* (figura 105) situados en una estructura *while*, cuya condición para salir de él es que el motor llegue hasta su posición final, para que sus posiciones se vayan actualizando con las datos procedentes del láser.

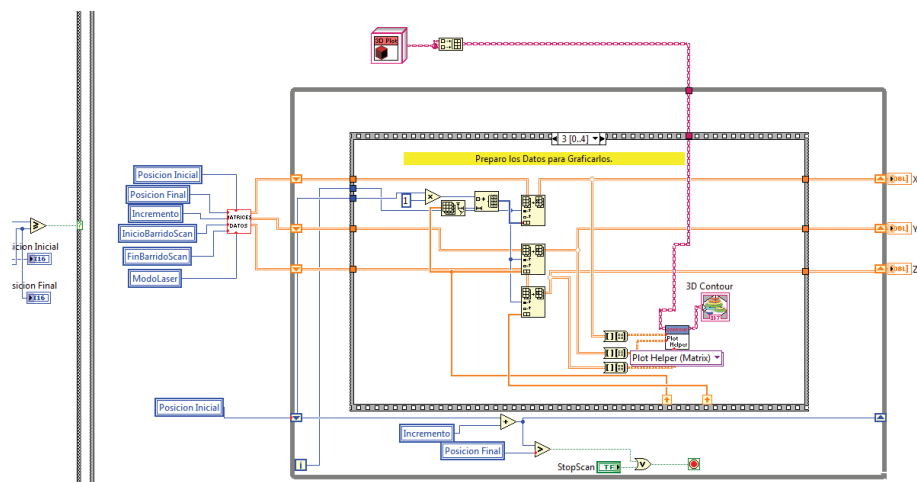


Figura 105: Shift register y diagrama de bloques donde se ha realizado el cambio

Este cambio realizado tampoco sirvió para solucionar el problema ya que unas veces fallaba y otras funcionaba correctamente.

#### 3.2.3.1.3 Compact Rio (cRIO)

Otra opción que propuesta por el soporte técnico de *National Instruments* fue probar el programa bajo un sistema *Compact Rio* (cRIO).

Un sistema Compact Rio, como su propio nombre indica, es un sistema compacto de salidas y/o entradas reconfigurables. Se constituye por un controlador (figura 107) basado en un sistema operativo en tiempo real, un chasis con una FPGA integrada y módulos de entrada/salida que se acoplan al chasis (figura 106). Gracias a esta arquitectura y a las posibilidades de conectar los cRIO a una LAN, se consigue tener un interfaz gráfico corriendo en un ordenador y la aplicación de control en un sistema determinista como lo son los equipos con SSOO en tiempo real.

La manera de crear el software que va a ejecutarse tanto en el controlador como en la FPGA, es a través de LabVIEW. Esto da la posibilidad de poder depurar a través del PC el software que corre dentro del cRIO, acortando considerablemente el tiempo de desarrollo. En la figura 108 se puede ver todo el sistema unido.

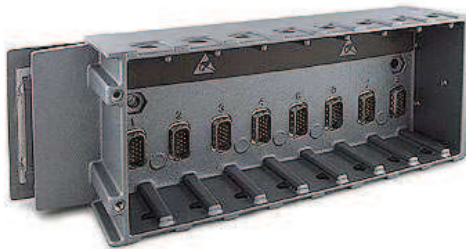


Figura 106: FPGA con chasis



Figura 107: Controlador



Figura 108: Controlador unido a chasis y FPGA

La diferencia de probar la aplicación bajo un sistema cRIO con respecto a un ordenador es que en este último caso el PC no solo se encarga de ejecutar nuestro programa sino que tiene otras muchas tareas de otros programas que atender. Ejecutar la aplicación bajo una plataforma cRIO proporcionaría un alto rendimiento y fiabilidad dado que usa un sistema operativo determinista en tiempo real frente a Windows que no es determinista.

Dado la falta de tiempo y la escasez de conocimiento en relación a los sistemas cRIO solo dio tiempo a gestionar el préstamo de los materiales necesarios para realizar las pruebas.

Los materiales prestados son:

- NI cRIO-9014

- Módulo NI 9870 serial con acceso a 4 puertos RS-232 diferentes. Se usará para conectar el motor y el láser.

Usar un sistema Compact Rio seguramente solucione este problema. Desde mi punto de vista no sería la mejor solución ya que adquirir un sistema como, por ejemplo, el que va a ser prestado es demasiado caro y la solución radica en la aplicación en el sistema donde se está ejecutando.

La solución propuesta por el soporte técnico en relación a que se están manejando datos muy grandes es un tanto errónea. Los *arrays* creados para almacenar los datos correspondientes a las coordenadas “X”, “Y” y “Z”, son *arrays* de dos dimensiones de elementos tipo *double*. El tamaño de los *arrays* viene determinado por las características del experimento que se vaya a realizar. En el peor de los casos, el motor tendría que recorrer los 2330mm que tiene la pasarela donde está situado y el mínimo incremento que se puede dar sería de 1 mm. Respecto al láser las características de barrido que podrían dar lugar a almacenar el mayor número de datos sería si se barre desde el ángulo 40 al 140 con una resolución de 0,25°.

El tamaño de los tres *arrays*, uno por coordenada, para el peor de los casos sería de aproximadamente 180Mb. Este dato se obtiene de multiplicar el número de posiciones que tiene un *array* por lo que ocupa cada una de las posiciones del mismo, 64 bit (*double*), y multiplicado por los tres *arrays* que se crean. El número de posiciones que tiene el *array* se obtiene de multiplicar las filas por las columnas del *array*, 2330 pasos que ha de dar el motor, filas, multiplicado por los ángulos barridos  $(140-40/0,25)$ , columnas.

Los dos ordenadores utilizados para probar la aplicación tienen memoria da datos suficiente para hacer frente a esta cantidad de datos, uno tiene 4GB de memoria RAM y otro 6GB, por lo que personalmente se descarta que el problema de la aplicación sea que se está utilizando una cantidad de datos muy grande.

### *3.2.3.2 Detección y corrección de errores coordenada "Z"*

A la vez que se trataba de solucionar el problema anterior se detectó y solucionó otro problema en relación con los datos obtenidos del láser, coordenada "Z", cuando se realizan barridos entre 90-135° para un experimento en particular. La solución adoptada ha sido procesar y corregir los datos correspondientes a la coordenada "Z" una vez se hayan obtenido.

Para ello se ha seguido el siguiente proceso.

Se han realizado varios barridos del suelo con las siguientes características:

-Láser:

- Amplitud: 90°-150°.
- Resolución: 0,5°.
- 50 muestras.

-Motor:

- Posición inicial: 10mm.
- Posición final: 30mm.
- Incremento: 5mm.

Utilizando un nivel se comprueba que la estructura del láser tiene una pequeñísima inclinación y que el suelo, situado bajo la estructura del láser, se encuentra a nivel. El resultado de los diversos barridos del suelo da como consecuencia el siguiente gráfico representado en la figura 109:



Figura 109: Desnivel del suelo

El eje vertical representa milímetros y el eje horizontal grados.

Como se puede apreciar en la figura 109 parece como si el suelo no fuese totalmente plano sino que tiene una pequeña inclinación.

Una vez obtenido estos datos se ha realizado una aproximación lineal de la gráfica para así procesar los datos y corregirlos.

Entre el ángulo 90 y 107 se ha tomado una recta plana, desde el ángulo 107,5 hasta el 126 una recta con una cierta pendiente, desde el ángulo 126,5 hasta el 130 una recta plana, desde el ángulo 130,5 hasta el 136 una recta con pendiente y finalmente, el último tramo, desde el ángulo 130,5 hasta el 150 una recta con una cierta pendiente.

En la figura 110 se ve como quedaría la gráfica:

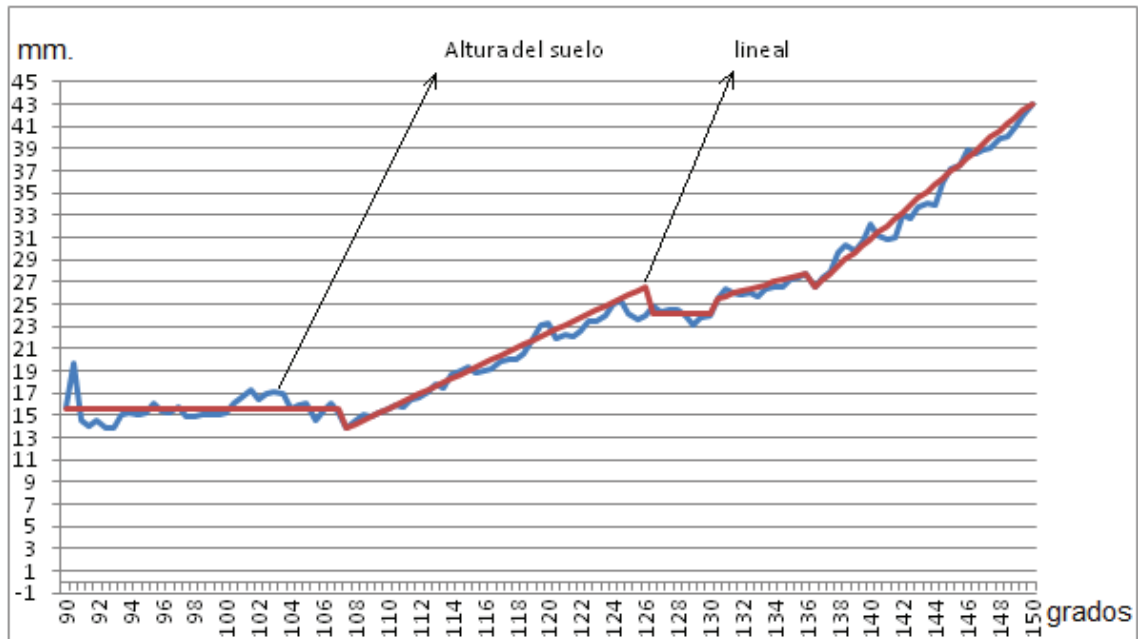


Figura 110: Desnivel con aproximación lineal

Desglose de la aproximación lineal:

-La primera parte de la gráfica lineal, la que va desde el ángulo 90 hasta el 107, se ha obtenido como promedio de los valores comprendidos entre estos ángulo, siendo este valor 15,5896756mm.

-La segunda parte es una recta con una pendiente. Esta recta va desde el ángulo 107,5 hasta el ángulo 126.

-La pendiente se ha obtenido realizando la siguiente operación:

$$m = 26,5710198 - 13,856404 / (126 - 107,5) = 0,68727653 \text{ mm/}^\circ.$$

Donde 26,5410198mm es la altura cuando estamos situados a 107,5°.

Donde 13,856404mm es la altura cuando estamos situados a 126°.

-La recta se calcula como:  $y = y_0 + m \cdot (x - x_0)$ . Donde:

$$y_0 = 13,856404 \text{ mm.}$$

$$m = 0,68727653 \text{ mm/}^\circ.$$

$$x = \text{grados actuales.}$$

$$x_0 = 107,5^\circ.$$

-La tercera parte de la gráfica, la que va desde el ángulo 126,5 hasta el 130 se ha obtenido como promedio de los valores comprendidos entre estos ángulo, siendo este valor 24,121833mm.

-La cuarta parte es otra recta con una cierta pendiente. Esta recta va desde el ángulo 130,5 hasta el ángulo 136.

-La pendiente se ha obtenido realizando la siguiente operación:

$$m=27,8146364-25,5640101/(136-130,5)=0,40920478\text{mm}/^\circ$$

Donde 27,8146364mm es la altura cuando estamos situados a 136°.

Donde 25,5640101mm es la altura cuando estamos situados a 130,5°.

-La recta se calcula como:  $y = y_0 + m \cdot (x - x_0)$ . Donde:

$$y_0 = 25,5640101\text{mm}.$$

$$m = 0,40920478\text{mm}/^\circ.$$

$$x = \text{grados actuales}$$

$$x_0 = 130,5^\circ.$$

-La quinta parte es, de nuevo, una recta con una cierta pendiente. Esta recta va desde el ángulo 136,5 hasta el ángulo 150.

-La pendiente se ha obtenido realizando la siguiente operación:

$$m=43,0656566-26,6140606/(150-136,5)=1,21863674\text{mm}/^\circ$$

Donde 43,0656566mm es la altura cuando estamos situados a 150°.

Donde 26,6140606mm es la altura cuando estamos situados a 136,5°.

-La recta se calcula como:  $y = y_0 + m \cdot (x - x_0)$ . Donde:

$$y_0 = 26,6140606\text{mm}.$$

$$m = 1,21863674\text{mm}/^\circ.$$

$$x = \text{grados actuales}$$

$$x_0 = 136,5^\circ.$$



Una vez realizada la aproximación lineal de la gráfica real se procede a restar a la gráfica real la aprox. lineal obteniéndose el siguiente resultado representado en la figura 111.

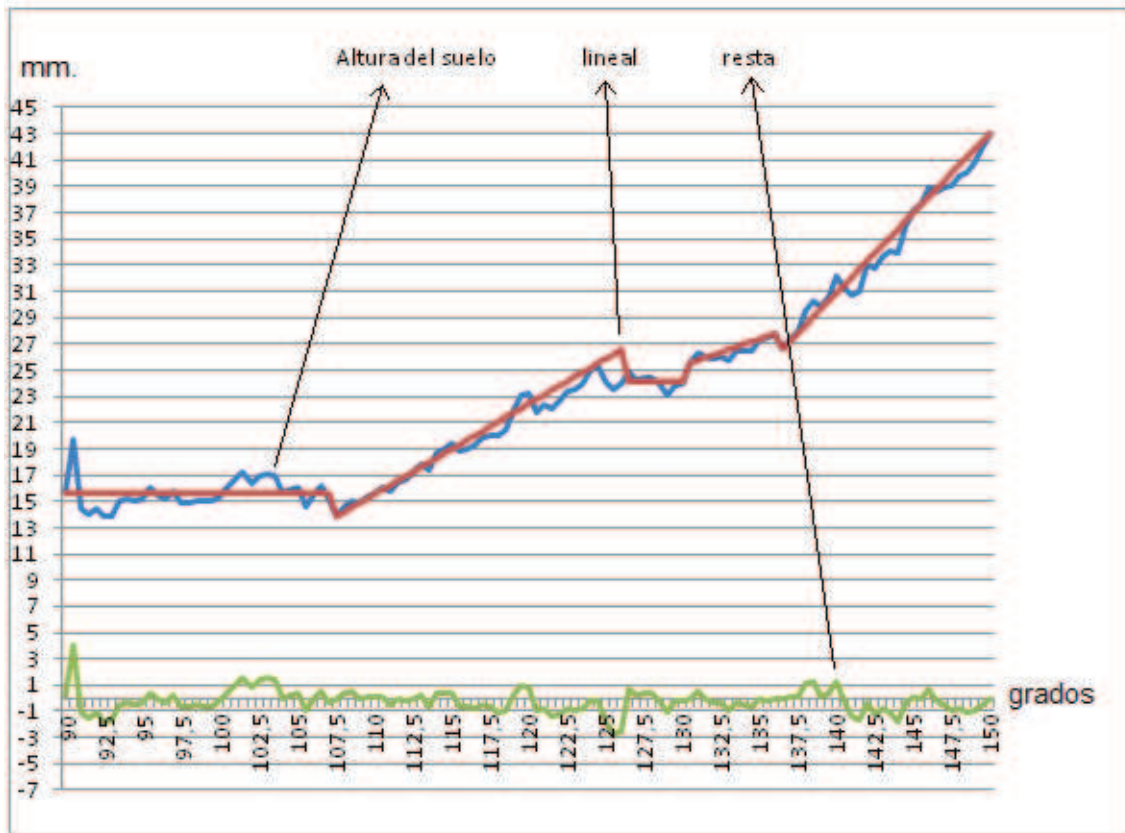


Figura 111: Corrección del error

Una vez obtenidos todos los datos de un barrido y copiados a un Excel hay que dirigirse abajo del todo, a la siguiente fila libre a partir de los datos copiados. En esta fila se pondrán los grados empezando por el 90 y acabando en el 150 con incrementos de 0,5 grados/columna. Se ponen ya que es necesario hacer referencia a los grados para realizar los tramos de la aprox. lineal que tienen pendiente.

En la siguiente fila se situarán los datos de la aproximación lineal, los datos de esta gráfica se obtienen realizando los pasos anteriores.

A continuación lo que queda es restar a la gráfica real su aprox. lineal consiguiendo así una representación más real.

## **4. CONCLUSIONES**

Partiendo de los objetivos descritos al principio del proyecto, primeramente se ha desarrollado una aplicación para recoger los datos provenientes de sensores de nivel, presión y caudal, y controlar una válvula. Para hacer posible que la aplicación pueda hacer frente a diferentes experimentos cuyo número de sensores pueda variar se ha utilizado un documento de Excel a través del cual se introducirán las características de los sensores utilizados y un control dentro de la aplicación para indicar el número de tarjetas que se vayan a utilizar.

De esta forma junto con la posibilidad de controlar una válvula desde la aplicación se podrá controlar, automatizar y estudiar de forma cómoda y sencilla cualquier experimento que requiera de la presencia de este tipo de sensores y de una válvula para controlar la cantidad de agua en el mismo, como podría ser el estudio de presas, canales, sumideros, etc.

El otro objetivo era detectar y corregir una serie de errores de una aplicación que controlaba dos dispositivos: un láser y un motor. Una vez detectado el fallo de la aplicación se realizaron una serie de correcciones relacionadas con temas de gestión de memoria sin conseguir que la aplicación funcionase correctamente. También se solicitó ayuda sobre el tema al soporte técnico de *National Instruments* proponiéndonos también una solución sobre temas de gestión de memoria que no solucionó el problema. Finalmente, siguiendo el consejo del soporte técnico, se inició un proceso de préstamo de un sistema Compact RIO para probar la aplicación sobre este sistema y ver si se solucionaba el problema.

Otro fallo que ha sido detectado y corregido está relacionado con la medición de la coordenada "X" que mide el láser. La estructura que sujeta al láser y al motor tiene una pequeña inclinación que hace que cuando se esté barriendo una superficie plana esta parezca estar inclinada. Para solucionar este error, dado que corregir esa pequeña inclinación es difícil, se ha optado por realizar una serie de operaciones a los datos obtenidos correspondientes a la coordenada "Z" para eliminar esa pequeña inclinación. La primera operación ha sido realizar una aproximación lineal de esa inclinación para que una vez se obtengan los datos de esa coordenada restarlos a la aproximación lineal y eliminar esa inclinación.

El objetivo de esta última aplicación es la de servir de apoyo para el estudio de presas construidas a escala y controlar posibles roturas en las mismas sacando imágenes en tres dimensiones a partir de los datos recibidos del láser. Dado que la aplicación no siempre fallaba se han podido realizar numerosos estudios, de esta

misma forma se podrán seguir realizando experimentos con la aplicación siempre que ésta no falle.

## **5. CARACTERÍSTICAS TÉCNICAS DE LOS EQUIPOS UTILIZADOS**

## 5.1 Módulo de entradas en corriente NI 9208

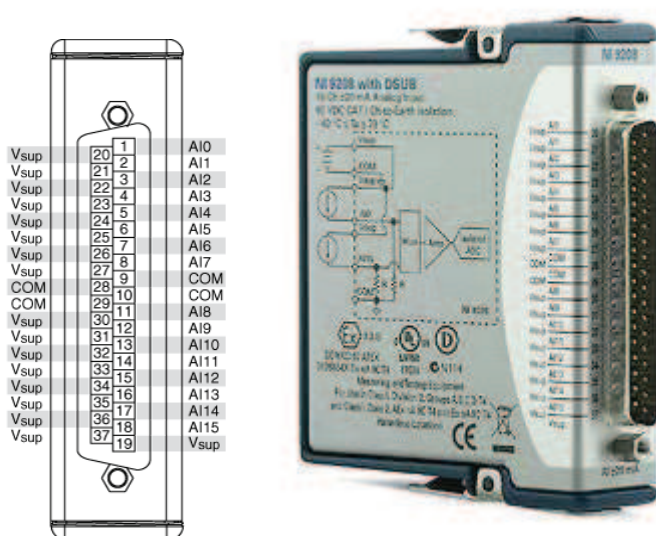


Figura 112: Tarjeta NI-9208

Fabricante.....*National Instruments*

General

Producto.....NI 9208

Familia de Productos.....E/S Industrial

Formato Físico.....*CompactDAQ, CompactRIO*

Número de Parte.....780968-01

Sistema Operativo/Objetivo.....Real-Time

Tipos de Medida.....Corriente

Compatibilidad con RoHS.....Sí

Acondicionamiento de Señales.....Entrada de corriente de 0-20 mA

Entrada Analógica

Canales.....0,16

Canales de un solo terminal.....16

Canales Diferenciales.....	0
Resolución.....	24bits
Tipo de ADC.....	Delta-Sigma
Velocidad de Muestreo.....	500S/s
Rendimiento (Todos los canales).....	500S/s
Rango de Corriente Máximo.....	-21.5 mA a 21.5 mA
Precisión Máxima del Rango de Corriente.....	0.161 mA
Rango de Corriente Mínimo.....	-21.5 mA a 21.5 mA
Precisión Mínima del Rango de Corriente.....	0.161 mA
Muestreo Simultáneo.....	No
Filtro Disponibles.....	60Hz, 60Hz
Impedancia de entrada.....	85Ω
Tiempo de conversión (por canal)	
Modo alta resolución.....	52 ms
Modo alta velocidad.....	2 ms
Pines Vsup	
Máxima Corriente.....	2 A
Máximo Voltage.....	30 V
Ruido de entrada	
Modo alta resolución.....	50nA <sub>rms</sub>
Modo alta velocidad.....	200nA <sub>rms</sub>
Estabilidad	
Variación de ganancia.....	22 ppm/°C
Variación del offset.....	62 nA/°C

#### Consumo de energía del chasis

Modo activo.....282 mW max

Modo "Sleep".....25  $\mu$ W max

#### Especificaciones Físicas

Longitud.....9 cm

Ancho.....9 cm

Conector de E/S.....D-Sub de 37 pines

Rango Temperatura de Operación.....-40 °C a 70 °C

Rango Temperatura de Almacenamiento.....-40 °C a 85 °C



5.2 Chasis NI CompactDAQ 9174 USB de 4 Ranuras



Figura 113: CompactDAQ 9174

Fabricante.....National Instruments

General

Producto.....cDAQ-9174

Formato Físico.....CompactDAQ

Tipo de Porducto.....Chasis

Número de Parte.....781157-01

Sistema Operativo/Objetivo.....Windows

Soporte para LabVIEW RT.....No

Humedad Relativa de Operación.....10%-90%

Chasis

Número de Ranuras.....4

Potencia Total Disponible.....15 W

Rango de entrada de voltaje.....9 V – 30 V

Disparo integrado.....No

## Contadores/temporizadores

Número de contadores/temporizadores.....4

Resolución.....32bits

## Interfaz de bus

Especificaciones de USB.....USB 2.0 de alta velocidad

Transferencias de datos de alto rendimiento.....7

Tipos disponibles.....E/S analógica,

E/S digital, entrada de contador/temporizador

## Impacto y vibración

Impacto operacional.....30 g

Rango aleatorio de frecuencia de operación.....5 Hz – 500 Hz

Vibración aleatoria.....0.3 g

## Especificaciones físicas

Longitud.....15.9 cm

Ancho.....8.81 cm

Altura.....5.89 cm

Peso.....574 g

Rango de temperatura.....-20°C a 55°C

Altitud máxima.....5000m

### 5.3 Fuente de Alimentación NI PS-15



Figura 114: Fuente de alimentación PS-15

Fabricante.....*National Instruments*

#### Especificaciones físicas

Ancho.....32 mm

Altura.....124 mm

Longitud.....117 mm

Peso.....500 g

Temperatura de funcionamiento.....-25°C a 70°C

Protección.....IP 20

#### Características eléctricas

Tensión de entrada.....90 V<sub>ac</sub> a 132 V<sub>ac</sub>

Frecuencia de entrada.....50Hz a 60Hz ±6%

Factor de potencia.....0.56 @ 120 V<sub>ac</sub>, 24 V, 5A

Tensión de salida.....24 V<sub>dc</sub> a 28 V<sub>dc</sub>

Potencia de salida.....120 W

Corriente de salida.....	5 A @24 V/ 4.3 A @28 V
Rizado y tensión de ruido.....	50 mV <sub>pp</sub>
Capacidad de salida.....	1800μF
Eficiencia.....	89.4% @120V <sub>ac</sub> , 24V, 5 A

## 5.4 SENSOR DE NIVEL ULTRASÓNICO MÓDELO LU-05



Figura 115: Sensor de nivel LU-05

Fabricante.....	FLOWLINE
Especificaciones	
Rango medición.....	5 cm a 1.2 m
Precisión.....	3mm
Resolución.....	0.5mm
Ancho del haz.....	5cm
Indicador LED.....	alimentación, calibración y diagnóstico
Memoria.....	No volátil
Alimentación.....	12Vdc a 28Vdc
Resistencia de bucle.....	500Ω @24Vdc
Señal de salida.....	4-20 mA, dos cables
Señal invertida.....	4-20mA o 20-4 mA
Temperatura de funcionamiento.....	-20°C a 60°C

Punto de ajuste de fábrica

4mA.....1.2m

20mA.....5cm

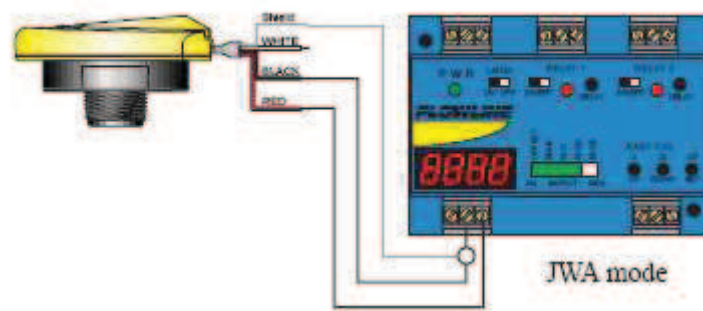
Conversión

Distancia cm	Corriente mA	Distancia cm	Corriente mA	Distancia cm	Corriente mA
5	20.0	50	13.7	100	6.8
10	19.3	60	12.3	110	5.4
20	17.9	70	11.0	120	4
30	16.5	80	9.6		
40	15.1	90	8.1		

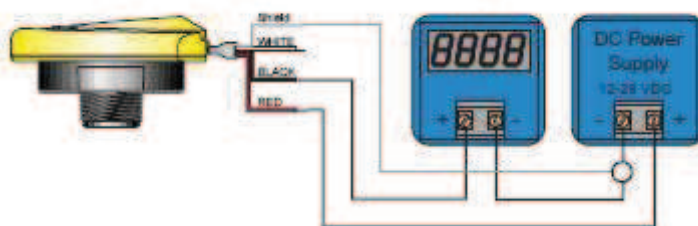
Tabla 10: Relación altura corriente

Cableado

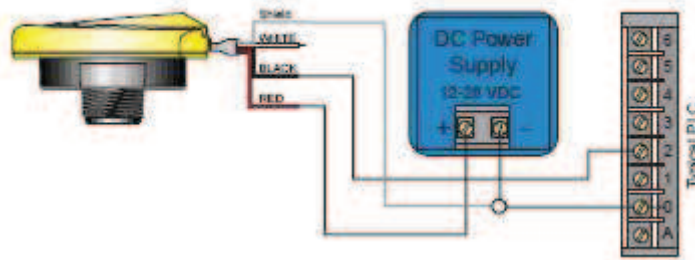
**A. Wiring to a FLOWLINE LC52-1001 Controller**



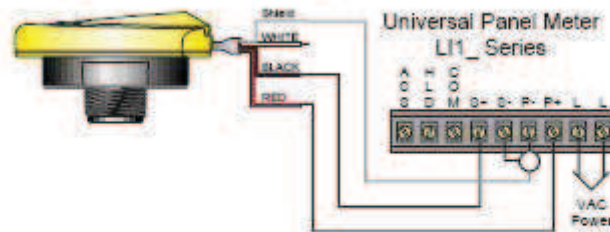
**B. Wiring to a Typical Two-Wire Loop Powered Indicator**



**C. Wiring to a Typical Programmable Logic Controller**



**D. Wiring to a Flowline LI1\_-1001 Universal Panel Meter**



**E. Wiring to a Flowline LI42-1001 MicroPoint Indicator**

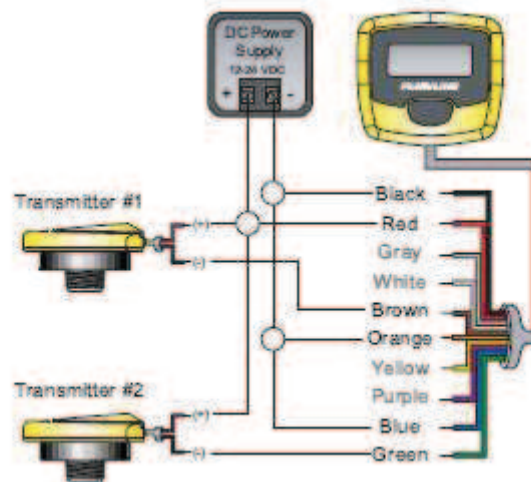


Figura 116: Cableado sensor LU-05

## Dimensiones

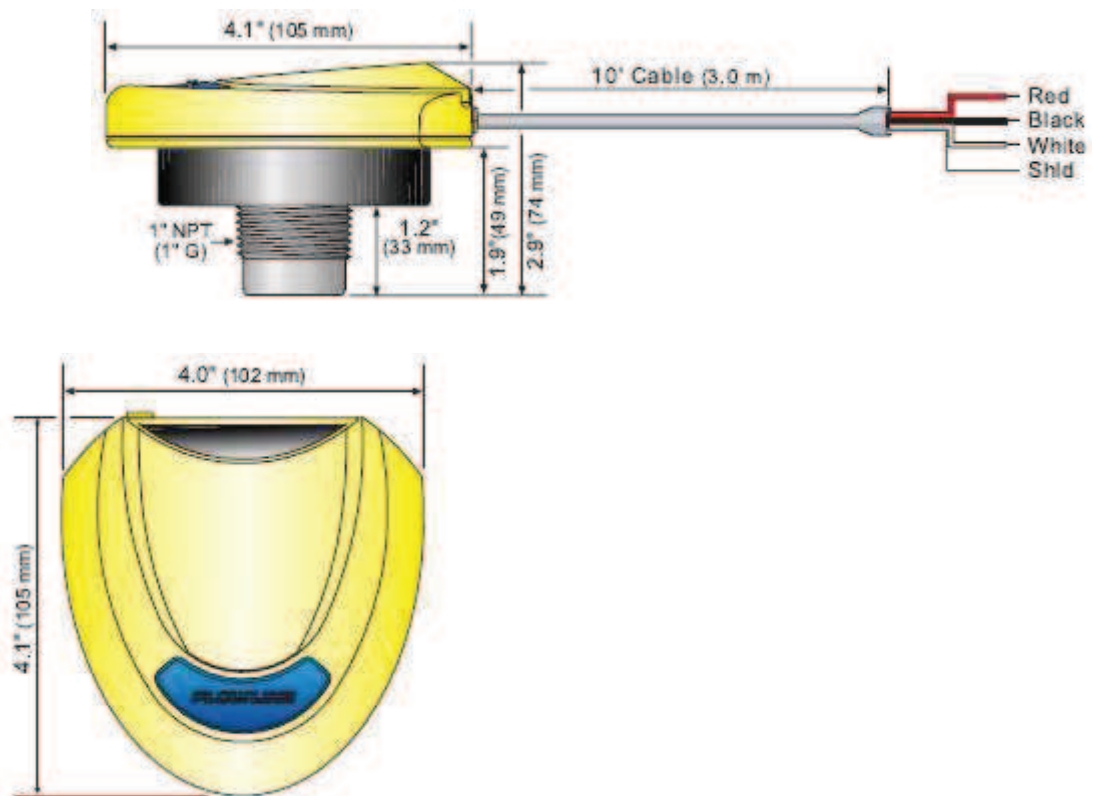


Figura 117: Dimensiones sensor LU-05



## 5.5 SENSOR DE PRESIÓN MODELO XA-300



Figura 118: Transmisor de presión XA-300

Fabricante.....Instrumentación Aplicada

Material en contacto

Rosca a proceso.....Acero inoxidable AISI.316.L

Sensor.....Cerámico de óxido de aluminio ( $Al_2O_3$  96%),

siendo la técnica utilizada la piezoresistiva

Junta tórica.....NBR

Datos técnicos

Presiones.....Relativas, absolutas, vacío

Rangos de medición.....De 0...0,250 Bar a 0...250 Bar

(rangos bajo demanda)

Resolución.....0,01 a 0,014% FE

Error combinado del sensor..... $\leq 0,3\%$  FE

(Linealidad, con histéresis y repetitibilidad)

Tensión de aislamiento del sensor.....2KV

Tiempo de respuesta.....Menor a 1mseg

Señal de salida normalizada.....	4-20mA 2 hilos-lineal
Tensión de alimentación.....	10Vdc-35Vdc
Máxima resistencia de carga.....	$R_a \leq [U_b(V_{dc}) - 10(V_{dc})] / 0,02 A_{dc}$
Protecciones eléctricas.....	Si, de polaridad y circuito

#### Características constructivas

Tipo de sensor.....	Cerámico
Material del cuerpo exterior.....	Acero Inoxidable
Grado de protección.....	IP-65 (EN60529)
Conexión eléctrica.....	Conector de tres polos
Temperatura	
Ambiente.....	-5°C a 90°C
Almacenamiento.....	-10°C a 80°C

#### Dimensiones (mm.)

##### Rosca a proceso

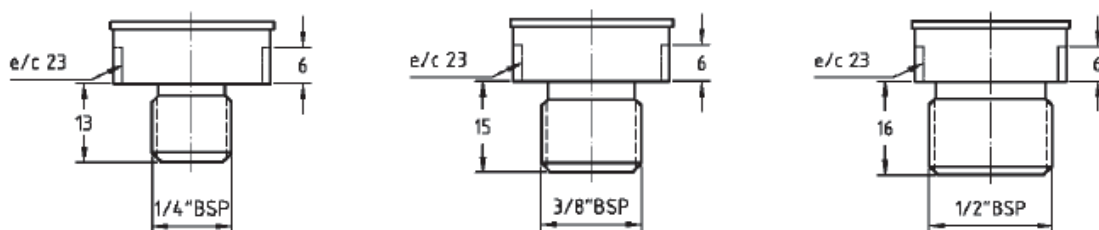


Figura 119: Rosca transmisor XA-300

## Cuerpo

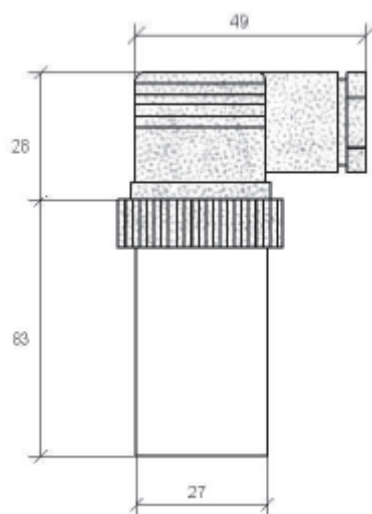


Figura 120: Dimensiones transmisor XA-300

## Conexiones eléctricas

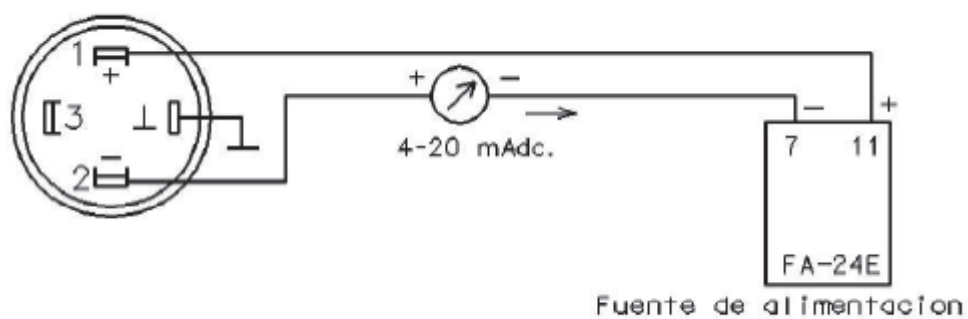


Figura 121: Conexiones transmisor XA-300

## 5.6 SENSOR DE PRESIÓN MODELO XA-700



Figura 122: Transmisor de presión XA-700

Fabricante.....	Instrumentación Aplicada
Material en contacto	
Rosca a proceso.....	Acero inoxidable AISI.316.L
Sensor.....	Cerámico de óxido de aluminio ( $Al_2O_3$ 96%), siendo la técnica utilizada la piezoresistiva
Junta tórica.....	NBR
Datos técnicos	
Presiones.....	Relativas, absolutas, vacío
Rangos de medición.....	De 0...0,250 Bar a 0...250 Bar (rangos bajo demanda)
Resolución.....	0,01 a 0,014% FE
Error combinado del sensor.....	$\leq 0,3\%$ FE (Linealidad, con histéresis y repetitibilidad)
Tensión de aislamiento del sensor.....	2KV
Tiempo de respuesta.....	Menor a 1mseg

Señal de salida normalizada.....	4-20mA 2 hilos-lineal
Tensión de alimentación.....	10Vdc-35Vdc
Máxima resistencia de carga.....	$R_a \leq [U_b(V_{dc}) - 10(V_{dc})] / 0,02 A_{dc}$
Protecciones eléctricas.....	Si, de polaridad y circuito

#### Características constructivas

Tipo de sensor.....	Cerámico
Material del cuerpo exterior.....	Acero Inoxidable

Grado de protección.....IP-68 (EN 60529)  
con cierre hermético permanente

Conexión eléctrica.....Mediante cable de tres polos  
(3x0,34 mm<sup>2</sup>)

#### Temperatura

Ambiente.....	-5°C a 90°C
Almacenamiento.....	-10°C a 80°C

#### Dimensiones (mm.)

Rosca a proceso

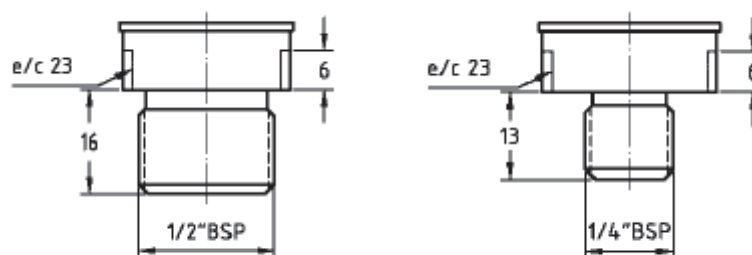


Figura 123: Rosca transmisor XA-700

## Cuerpo

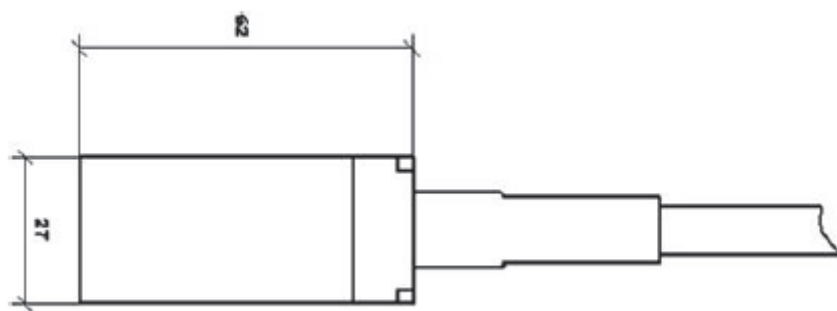


Figura 124: Cuerpo transmisor XA-700

## Conexiones eléctricas

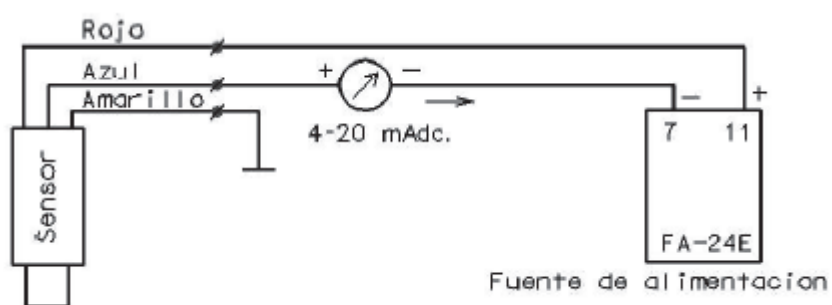


Figura 125: Conexiones eléctricas transmisor XA-700

## 5.7 VÁLVULA AUMA Matic AM 01.1 / AM 02.1



Figura 126: Válvula AUMA

Fabricante.....	AUMA
Tipo de servicio	
SA.....	Reducido S2 – 15 MIN
SAR.....	Intermitente S4 – 25%
Motores.....	Motor trifásico asíncrono, tipo IM B9 según IEC 34
Clase de aislamiento.....	F, tropicalizado
Protección del motor.....	Termostatos (NC)
Auto-bloqueo.....	Sí; para velocidad 4-90rpm
Finales de carrera.....	Interruptor sencillo (1 NC + 1 NA) para cada sentido
Señal analógica de posición.....	Potenciometro o 0/4 - 20 mA
Indicador mecánico de posición.....	Indicación continua, disco indicador ajustable con símbolo ABIERTO y CERRADO
Indicación de marcha.....	Intermitente (estándar par SA, opción para SAR)

Mando manual.....	Para ajustes y operación de emergencia, el volante no gira durante la operación eléctrica
Conexión al control.....	Conector múltiple AUMA con terminales para atornillar
Alimentación.....	ver placa de características en control
Control.....	Entradas digitales 24 Vcc, ABRIR – PARAR – CERRAR.
Consumo aprox.10mA por entrada.	
Salidas digitales.....	5 relés de salida con contactos recubiertos de oro
Salida de tensión.....	Tensión auxiliar 24 Vcc, máx. 50mA para alimentación de las entradas digitales, aislada galvánicamente de la alimentación interna.
Mandos locales.....	Selector LOCAL – OFF – REMOTO (bloqueable en las tres posiciones) Pulsadores ABRIR – PARAR – CERRAR
Grado de protección ambiental.....	IP 67
Temperatura ambiente.....	-25 °C hasta +70 °C
Posicionador electrónico	
Señal de consigna.....	0/4 – 20mA (opción 0 – 5V)
Señal de salida.....	0/4 – 20mA (opción 0 – 5V)
Sensibilidad (banda muerta).....	0,5 % - 2,5 %
Ajuste fino “sens” (P 7)	
(solo útil para velocidades en actuador <16 l/min)....	Min 0,25 %
Tiempo de pausa “toff2 (P 8).....	0,5 – 15s



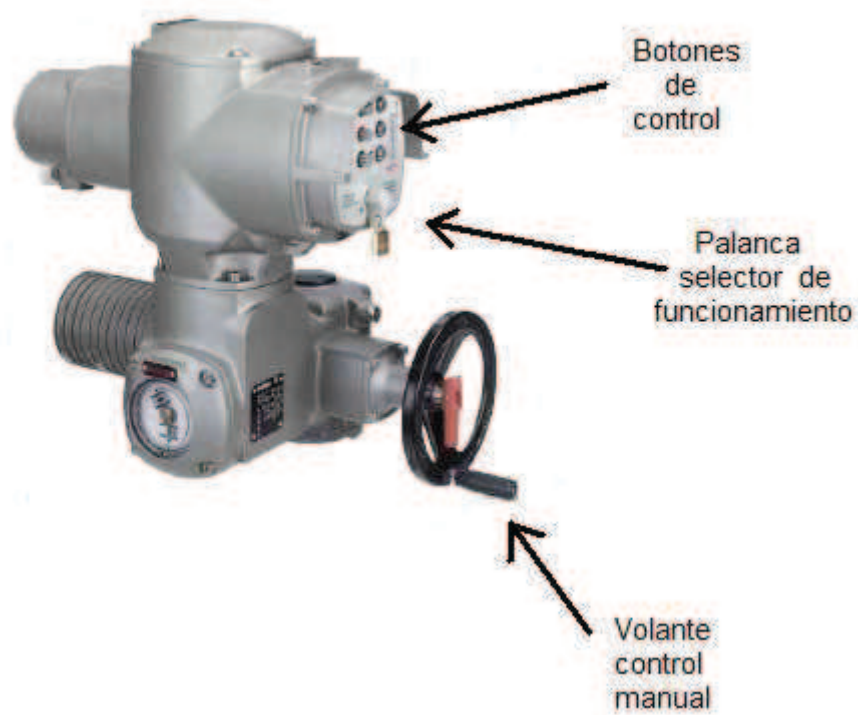


Figura 127: Controles válvula

## **6. BIBLIOGRAFÍA**

- Mariano Ruiz y Guillermo de Arcas. *INTRODUCCIÓN A LabVIEW*. Dpto. de Sistemas Electrónicos y de Control, Madrid 2006.
- National Instruments. *LabVIEW™ Core 2 Course Manual*. Octubre 2009.
- National Instruments. *Labview™ Intermediate I: Successful Development Practices Course Manual*. Abril 2008
- National Instruments. *LabVIEW™ Basics I: Introduction Course Manual*. Octubre 2008.
- Páginas web de:
  - National Instruments  
<http://www.ni.com/es/>
  - INSTRUMENTACION APLICADA S.A.  
[http://www.inst-apli.com/CATALOGOS/www.inst-apli.com\\_TRANSMISOR\\_PRESION\\_mod.XA-300.pdf](http://www.inst-apli.com/CATALOGOS/www.inst-apli.com_TRANSMISOR_PRESION_mod.XA-300.pdf)  
[http://www.inst-apli.com/CATALOGOS/www.inst-apli.com\\_TRANSMISOR\\_PRESION\\_mod.XA-700.pdf](http://www.inst-apli.com/CATALOGOS/www.inst-apli.com_TRANSMISOR_PRESION_mod.XA-700.pdf)
  - FLOWLINE  
<http://www.flowline.com/pdf/discontinued/echosonic-lu05-lu11-lu12-lu13-m.pdf>
  - AUMA  
[http://www.auma.com/uploads/media/sp\\_import2/betriebsanleitung\\_en/antriebe\\_mit\\_steuerung/drehantriebe\\_mit\\_auma\\_matic/ba\\_sar1am1\\_es.pdf](http://www.auma.com/uploads/media/sp_import2/betriebsanleitung_en/antriebe_mit_steuerung/drehantriebe_mit_auma_matic/ba_sar1am1_es.pdf)